AFOSR-TR-97-0440

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704 0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
April 1994

3. REPORT TYPE AND DATES COVERED
Final, Sep 1993-Feb 1994
1 Sep 93- 31 August, 96

4. TITLE AND SUBTITLE
Automatic Target Detection and Recognition:
A Wavelet Based Approach

5. FUNDING NUMBERS
F49620-93-1-0490

6. AUTHOR(S)
Drs. A.J. Devaney, R. Raghavan, H. Lev-Ari
E. Manolakos, and M. Kokar

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Northeastern University, Boston, MA 02115

8. PERFORMING ORGANIZATION REPORT NUMBER
NU-ARPA1

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Office of Scientific Research
110 Duncan Avenue, Suite B115
Bolling AFB, DC 20332

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Distribution Unlimited

N CODE

13. ABSTRACT *(Maximum 200 words)*

Wavelet based target detection and identification algorithms for radar applications are presented and tested and evaluated on computer simulated data. The algorithms make use of a *scale sequential* and/or *scale recursive* paradigm where computations are performed within and across scales in a multiresolution analysis (MRA) of the sensor data relative to a compactly supported discrete orthonormal wavelet basis. It is argued that such procedures are computationally efficient and offer promise of yielding near optimal performance with a minimum CPU time burden. Specific applications considered in the report include automatic target identification from high range resolution radar (HRR), target detection in the presence of fractal noise and the integration of multisensor data in the tracking of aircraft. Other applications addressed include scale recursive optimal filtering and the synthesis of parallel architectures for the 1-D discrete wavelet transform. The report includes a full discussion of the theory behind the various detection and identification algorithms plus results from Monte Carlo simulations.

14. SUBJECT TERMS
Wavelets, Radar,
Automatic Target Idenfication, Target Recognition,
Multiresolution Analysis

15. NUMBER OF PAGES
72

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

FINAL REPORT
Automatic Target Detection And Recognition: A Wavelet Based Approach ARPA Grant F49620-93-1-0490

A. J. Devaney
R. S. Raghavan
H. Lev-Ari
E. S. Manolakos
M. Kokar
Department of Electrical and Computer Engineering
Northeastern University
Boston, MA 02115

January 25, 1997

Objectives and Status

This project had a start date of September 1, 1993 and ended on 31 August, 1996. The funding levels were $ 148,495 for the six month period 9/1/93-2/28/94, $ 298,515 for 3/1/94-2/28/95, $ 303,987 for 3/1/95-2/28/96, and funding of $ 154,905 for the final six months 3/1/96-9/1/96. The PI for the project is Professor Tony Devaney and the other principles are Professors Ram Raghavan, Hanoch Lev-Ari, Elias Manolakos, and Mitch Kokar all from the Department of Electrical and Computer Engineering at Northeastern University

The overall objective of the project is to develop, test and evaluate wavelet based algorithms for automatic target detection and identification. The work and results obtained can be roughly broken down into three sub-areas:

- Wavelet based scale recursive and scale sequential algorithms for HRR and SAR,
- Algorithms and Architectures for fast and efficient computation of Discrete Wavelet Transforms,

- Wavelet based sensor fusion algorithms.

Major Accomplishments

- ATR scale sequential and recursive algorithms have been developed for high range resolution radar (HRR) that operate at near optimum performance levels with computation times orders of magnitude below conventional fine scale maximum likelihood based algorithms. These algorithms have been extended to the detection of electronic signals in ELINT operations. Professor Devaney has presented this work at a number of ARPA sponsored workshops as well as at other meetings such as at a special workshop on ATR held at Phillips Laboratory in Albuquerque in the summer of 1995. He also presented a three hour tutorial on this and related work at the National Security Agency in February 1996.

- Multi-resolution based ATR algorithm has been generalized to cases where a target to be detected/recognized is described only in terms of the various scales that make up the target signature. Knowledge of the actual target signature within each scale is not required in this formulation. The approach has several features that make it attractive for the ATD/R application: (i) Test for target presence in each scale associated with the signal subspace is carried out independently, (ii) the false alarm probability for each scale may be specified independently and (iii) the approach offers a greater degree of protection against missing a target since the exact signature of the target is not required in the algorithm. Professor Raghavan has presented this and related work at Lincoln Laboratories in March of 1996 and at Rome Laboratories at Hanscom AFB in September 1995.

- An analytical framework for optimal design of MSE (Wiener) filters in the multiresolution (subband) domain has been developed that exhibits excellent cost-performance tradeoff, and is capable of achieving significant reduction in implementation cost with only a minor degradation in performance. The design is based on the notions of: (i) scale-recursiveness, (i) sparse estimation, and (ii) optimized front-end tree-structured filter bank configurations and it has been shown that the key to efficient cost-performance tradeoff when using sparse estimation is the reduction of overlap between adjacent channels of the filter bank. This can be achieved by a suitable selection of the prototype filter bank response.

- M. Kokar and students developed a feature selection method based on model theory have implemented this scheme in the Automatic Feature Based Recognition System (AFBRS). The method has been tested on simulated ATR dataand the performance was compared with the recognition method described by Coifman and Saito. In these tests the method showed better recognition accuracy without introducing additional computational cost. We have also obtained reduced dimensionality of input data through model-theory based feature selection.

- The *synthesis* of parallel computational structures for the Discrete Wavelet Transform (DWT), starting from a thorough and systematic algorithmic data-dependence analysis was investigated by Prof. Elias S. Manolakos and graduate students. Improving algorithm performance through parallelism has been addressed from two points of view: By optimizing aggregate throughput in a series of problem instances by means of modular VLSI *architectures* having a small number of processing elements (PEs); and by means of *scalable algorithms* for parallel supercomputers having a large number of PEs and a fixed interconnection network, where the object is to minimize execution time of a single, potentially very large problem instance typical of SAR image processing. A family of modular application specific VLSI architectures was derived for the DWT as well as a family of parallel DWT algorithms for Linear, Mesh, and Hypercube processor networks. All parallel algorithms have been evaluated in terms of their theoretical and practical *scalability* properties.

- Model-Based Target Recognition was investigated within the framework of parameter estimation of hierarchical mixture densities by Professor Elias S. Manolakos and graduate students. An appropriately modified version of the Expectation-Maximization (EM) algorithm was introduced that generates estimates of the posterior probability of target presence in an image. The EM was extended to multiple level mixture hierarchies in order to deal with target translation and scale invariance. The approach can account for the non-stationary statistics of the input image as well as different noise models, two aspects that are very important for ATR. Our simulation results suggest that the scheme works very well even in the case of partially occluded targets presented at scales for which templates are not available.

- A large number of journal and conference papers have been produced detailing the work performed on this project. In addition, five M.S. theses have been produced and another M.S, and three doctoral theses are in progress.

Publications

- A.J. Devaney and K. Riley, "Scale sequential processing of HRR data for target recognition", in preparation.

- K. Riley and A.J. Devaney, "Wavelet processing of images for target detection", *International Journal of Imaging Systems and Technology* **7**, p.404-420, 1996.

- A.J. Devaney and B. Hiscomez, "Wavelet signal processing for radar target identification: a scale sequential approach," in **Wavelet Applications**, Harold Szu, editor, *Proc. SPIE 2242*, 389-399, 1994.

- R. S. Raghavan et.al, "Performance of the GLRT for Adaptive Vector Subspace Detection", *IEEE Trans. on Aerospace and Electronic Systems,* 1996 (to appear).

- R. S. Raghavan et.al, "Adaptive detection of signals in spherically invariant noise,"*IEEE Trans. on Signal Processing,* 1996, (in Review)

- J. Fridman, E.S. Manolakos, "Discrete Wavelet Transform on a SIMD massively parallel platform", Proc. of the *International Conference on Signal Processing Applications and Technology,* (ICSPAT'95), Boston, 1995.

- J. Fridman, E.S. Manolakos, "On the Scalability of 2-D Discrete Wavelet Transform Algorithms", *Multidimensional Systems and Signal Processing,* (special issue on wavelets and multiresolution signal processing), to appear 1996.

- H. Liu and H. Lev-Ari, “Scale-Recursive Optimal Filtering,” CDSP Technical Report No. TR-CDSP-94-25, Northeastern University, Dec. 1994.

- H. Liu and H. Lev-Ari, “Optimal Filtering in the Subband-Domain,” *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Detroit, MI, May 1995.

- Z. Korona and M.M. Kokar, “Target recognition using wavelet-based features,” in *Proc. SPIE AeroSense Conference: Wavelets,* Orlando, FL, April 1996.

- Z. Korona and M.M. Kokar, “Model-based fusion for multisensor target recognition,” in *Proc. SPIE AeroSense Conference: Image Exploitation and Target Recognition,* Orlando, FL, April 1996.

Contents

1. Devaney–Scale Sequential HRR

In this Chapter, M-ary hypothesis testing in the wavelet domain is detailed. The application of interest is target identification from large data base HRR radar where the hypotheses are the set of targets with various orientations and the signal used for classification is the measured HRR return. Decisions are made in a sequential fashion by starting at a coarse wavelet scale and proceeding to finer scales. As the wavelet scale is increased, hypotheses are ruled out according to a predefined decision criterion. Such an approach is computationally efficient and well suited to detection problems with large signature data bases.

The following sections present two forms of a scale sequential detection algorithm. The first form is optimal in the sense that it achieves the same error rate performance as the matched filter detector implemented at finest scale but with significant savings in computational time. The second form is a sub-optimal algorithm that is capable of achieving even greater computational savings with only a marginal decrease in error rate performance.

1..1 Hypothesis Testing in the Wavelet Domain

In order to perform hypothesis testing in the wavelet domain, a wavelet based interpretation of the log likelihood function is required. To develop such a quantity, consider the conventional log likelihood function

$$L(k) = \int dt\ [r(t) - s_k(t)]^2, \tag{1}$$

where $r(t)$ is a measured signal (e.g., a HRR return) and $s_k(t)$ a hypothesis (e.g., member of the signature data base). For a sufficiently large wavelet scale J (high Nyquist sampling rate), the Nyquist sample values $r(n)$ and $s_k(n)$ become equal to the scaling coefficients at scale J. We can thus approxmate the measured return and data base signal in a scaling function expansion of the general form

$$r(t) = \sum_n a_n^J\ \phi_n^J(t) = \sum_n r(n)\ \phi_n^J(t) \tag{2}$$

where ϕ_n^J are the scaling functions at scale J. Using the above representation for both $r(t)$ and $s_k(t)$ in Eq.(1), the log likelihood ratio becomes

$$L(k) = \int dt\ [\sum_n r_n^J\ \phi_n^J(t) - \sum_n s_n^J(k)\ \phi_n^J(t)]^2, \tag{3}$$

which, after making use of the orthonomality property of the scaling functions reduces to

$$L(k) = L^J(k) = \sum_n [r_n^J - s_n^J(k)]^2. \tag{4}$$

In Eq.(4) the log likelihood function has been completely expressed in terms of wavelet scaling coefficients at scale J. Using a similar approach, the log likelihood function can be expressed as a function of the scaling coefficients and wavelet coefficients at scale $J-1$. In particular, we find that

$$\begin{aligned} L(k) = L^J(k) &= \sum_n \left[r_n^{J-1} - s_n^{J-1}(k)\right]^2 + \sum_n \left[wr_n^{J-1} - ws_n^{J-1}(k)\right]^2 \\ &= L^{J-1}(k) + wL^{J-1}(k). \end{aligned} \tag{5}$$

By induction, the log likelihood function can be decomposed to an arbitrary wavelet scale J_o and written as

$$L^J(k) = \sum_{j'=J_o}^{J-1} wL^{j'}(k) + L^{J_o}(k) \tag{6}$$

where

$$L^{J_o}(k) = \sum_n [r_n^{J_o} - s_n^{J_o}(k)]^2 \tag{7}$$

$$wL^{j'}(k) = \sum_n [wr_n^{j'} - ws_n^{j'}(k)]^2 \tag{8}$$

The above expression for the log likelihood function will be used to perform hypothesis testing in the wavelet domain. The following section details a computationally efficient means for computing Eq(6) for large signature databases.

1..2 Scale Sequential Hypothesis Testing

The motivation for interpreting the M-ary detection problem in the wavelet domain is to reduce the computational cost of making a decision. For a large signature database, an optimal algorithm which computes Eq.(1) for every hypothesis will exact a large computational cost. Such a cost will reduce the rate at which decisions can be made. An attractive feature of Eq.(8) is that the range of the summation decreases geometrically with decreasing scale j. This feature can be exploited by initially testing hypotheses at a low wavelet scale and sequentially proceeding to higher scales according to a decision criterion. By ruling out unlikely hypotheses at low wavelet scales, the "full" log likelihood function (i.e. $L^J(k)$) is computed over a subset of the signature database and a computational savings is realized.

In short, the scale sequential detection scheme can be described as follows. For a signature database with K hypotheses, Eq.(7) is computed for the entire signature database. A decision criterion is employed to rule out hypotheses from consideration at the scale J_o+1. The test continues to scale $J_o + 1$ where Eqs.(6), (7), and (8) are used to compute $L^{J_o+1}(k)$ for only those hypotheses that tested positive at scale J_o. Similarly, the test proceeds in a sequential fashion to higher wavelet scales. Once the finest scale J is reached, the full log likelihood function is computed over a subset of the signature database resulting in reduced computational cost and increased decision rate. The following section presents the decision criterion that has been used to leave behind hypotheses while sequentially proceeding up wavelet scales.

1..3 Scale Sequential Testing using the Stack Algorithm

As described in the previous section, once the likelihood functions are computed at a given scale, a decision criterion is used to rule out or leave behind hypotheses from consideration at higher scales. This section details the use of probabilistic decoding as a decision criterion.

The scale recursive detection problem can be viewed in a probabilistic decoding framework where the encoding tree has K branches (K = number of hypotheses) stemming from the root node and a single branch from all successive nodes. The depth of the tree is equal to the number of levels of wavelet decomposition. The encoding tree for the scale recursive detection problem is shown in figure 1.

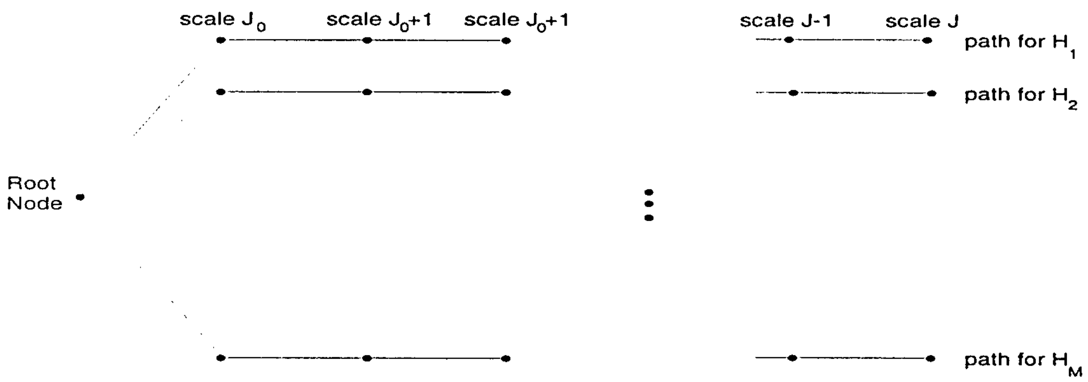

Figure 1: Encoding tree for the scale recursive detection problem

The stack decoder forms a cumulative metric from a sum of branch metrics. This is entirely analogous to the scale sequential detection problem where the likelihood function is built up by adding the contributions from the wavelet coefficients at each scale. The scale sequential detection problem can now be summarized as follows:

1. Compute the wavelet decomposition of the received signal.
2. Compute $L^{J_o}(k)$ for each hypothesis and form a sorted stack.
3. Using (6) compute $L^{j+1}(k)$ for the hypothesis having the smallest $L^j(k)$ and replace $L^j(k)$ with $L^{j+1}(k)$ in the stack.
4. Repeat step 3 until a hypothesis reaches the finest scale and is minimum. This hypothesis is chosen as the correct hypothesis.

With the algorithm detailed, the choice of stack metric can be addressed. In this report, two stack metrics are considered. These two metrics will be denoted $L2$ and $L2_{norm}$ and are stated below for a particular scale and hypothesis:

$$L2^j(k) = \sum_{n=1}^{N_j}[r^j(n) - s^j_k(n)]^2 \tag{9}$$

$$L2^j_{norm}(k) = \frac{1}{\sqrt{N_j}}\left[\sum_{n=1}^{N_j}(r^j(n) - s^j_k(n))^2 - N_j\sigma^2\right] \tag{10}$$

where $N_j = 2^j$ and σ^2 is the variance of the additive white Gaussian noise (AWGN) in the received signal.

The $L2$ metric is optimal with respect to error rate performance. This can be explained by noting that $L2^j$ is non-decreasing with increasing j. Consequently if $L2^J(k)$ is, for some hypothesis k, minimum compared with $L2^j(k')$ for $j \leq J$ and all k' it is also minimum with respect to all k' at the finest scale J; i.e., it remain minimum if all hypotheses are propagated to the finest scale. Testing all hypotheses at the finest scale corresponds to optimum M-ary detection which leads to the conclusion the stack algorithm with the $L2$ metric is guaranteed to arrive at the same decision as the optimal test.

The statistics of the $L2$ metric under both the correct and incorrect hypotheses are:

correct hypothesis:

$$\text{mean} = N_j\sigma^2 \tag{11}$$

$$\text{variance} = 2N_j\sigma^4$$

incorrect hypothesis: (k=correct hypoth. and l=hypoth. under test)

$$\text{mean} = N_j\sigma^2 + \frac{1}{\sqrt{N_j}}D_{l,k} \tag{12}$$

$$\text{variance} = 2N_j\sigma^4 + 4\sigma^2 D_{l,k}$$

where $D_{l,k}$ is the Euclidean distance measure defined as

$$D_{l,k} = \sum_{n=1}^{N_j}[s_l^j(n) - s_k^j(n)]^2. \tag{13}$$

From the statistics of the $L2$ metric it is apparent that as a hypothesis (either correct or incorrect) is propagated to higher scales, the mean and variance will tend to increase with N_j. As previously stated, the algorithm will propagate the smallest cumulative metric. If the algorithm is propagating the correct hypothesis, on average the cumulative metric will increase and it is likely that the correct hypothesis will not remain at the top of the stack. This results in the algorithm propagating incorrect hypotheses (whose metrics will also tend to increase) before returning to the correct hypothesis. Although the algorithm will tend to propagate incorrect hypotheses with the $L2$ metric, under most circumstances all incorrect hypotheses will not be propagated to the finest scale. This results in significant computational savings while maintaining optimality.

The $L2_{norm}$ metric is sub-optimal, but it is considered because it significantly increases the computational savings over the $L2$ metric while sacrificing a small amount of error rate performance. The motivation behind the $L2_{norm}$ metric is to have a stack metric that remains small when propagating the correct hypothesis and tends to grow when propagating an incorrect hypothesis. By defining the $L2_{norm}$ metric as a zero-mean, normalized version of the $L2$ metric along the correct path these requirements are satisfied. The statistics of the $L2_{norm}$ metric are stated below

correct hypothesis:

$$\text{mean} = 0 \tag{14}$$

$$\text{variance} = 2\sigma^4$$

incorrect hypothesis: (k=correct hypoth. and l=hypoth. under test)

$$\text{mean} = \frac{1}{\sqrt{N_j}} D_{l,k} \tag{15}$$

$$\text{variance} = 2\sigma^4 + \frac{4\sigma^2}{N_j} D_{l,k}$$

where $D_{k,l}$ is defined as before. As required, under the correct hypothesis the $L2_{norm}$ metric has a zero mean and constant variance. This results in the cumulative for the correct hypothesis remaining close to zero (on average) and therefore at or near the top of the stack. Conversely, under the incorrect hypothesis, the $L2_{norm}$ metric has a mean which will diverge from zero. The variance will also diverge but not as fast as the mean since the $D_{k,l}$ term varies as $1/N_j$ as opposed to $1/\sqrt{N_j}$ in the expression for the mean. As a result, the algorithm will quickly dismiss an incorrect hypothesis from the top of the stack. This facilitates the search for the correct hypothesis and once it is found the algorithm will quickly propagate it to the finest scale and finish. This means that fewer hypotheses are propagated than would be with the $L2$ metric resulting in further computational savings.

1..4 Complexity and Implementation

The detection schemes that the stack algorithm will be compared to in the following section are strictly arithmetic. The stack algorithm on the other hand uses both arithmetic and comparison operations. The arithmetic operations come from computing the stack metric while the comparisons come from forming and updating the stack (i.e. sorting the stack and placing new stack metrics in their proper locations). In order to develop a fair measure of complexity for the stack algorithm, both of these operations must be considered. In this paper, the measure of arithmetic complexity is taken to be FLOPs/decision. A single FLOP is assigned for each addition and multiplication. The comparisons that are performed during the execution of the algorithm are counted by assigning a one FLOP penalty for each comparison. This penalty can be justified by noting that a simple comparison between two numbers could involve taking their difference (one FLOP) and checking for positivity. With the measure of complexity defined, the implementation and computational analysis of the algorithm can be addressed.

The method of organizing and updating the stack can strongly affect the complexity of the algorithm. The approach taken in this paper is to first sort the stack (low to high) at the coarsest scale. Since the stack metrics are computed in a serial fashion, the sort consists of inserting each new metric into its proper position as the stack is built up. The insertions were made using a binary insertion algorithm which on average requires $\log_2 m$ comparisons where m is the size of the array on which the insertion is performed. From this follows that the stack sort at the coarsest scale requires

$$\sum_{m=2}^{K} \log_2 m \ \text{ comparisons} \tag{16}$$

where K is the total number of hypotheses to be tested. As previously stated, when propagating a hypothesis, the new stack metric (computed at the next higher wavelet scale) is placed into the stack in the proper location. This was also accomplished using a binary insertion algorithm thus requiring $\log_2 K$ comparisons for each propagation.

Evaluating the number of computations required to compute the stack metric is straightforward. The $L2$ metric requires N_j subtractions, N_j multiplications, and $N_j - 1$ additions resulting in the following expression

$$3N_j - 1 \ \text{ FLOPs/hypothesis.} \tag{17}$$

Similarly the $L2_{norm}$ metric requires $3N_j + 1$ FLOPs/hypothesis. Since $3N_j$ is in most cases the dominant term, (17) is used for both metrics.

With the complexity of maintaining the stack and computing the stack metric detailed, a lower bound on computational complexity can be derived. The lower bound will correspond to the case where the algorithm immediately propagates the correct hypothesis to the finest scale and finishes (i.e. no incorrect hypotheses are propagated). Using (16) and (17) this lower bound can be written as

$$T_s = K[3 \cdot 2^{J_o} - 1] + \sum_{m=2}^{K} \log_2 m + \sum_{m=J_o}^{J-1} [3 \cdot 2^m - 1] + (J - J_o) \log_2 K \tag{18}$$

where

$$\begin{aligned} J_o &= \text{the coarsest wavelet scale} \\ J &= \text{the finest wavelet scale} \\ K &= \text{the total number of hypotheses.} \end{aligned}$$

The first two terms on the right hand side of (18) reflect the cost of computing the stack metrics at the coarsest scale and forming the stack itself. The last two terms on the right hand side of (18) reflect the cost of immediately propagating the correct hypothesis to the end and finishing. The variability of the computational effort makes it difficult to derive a closed-form expression for the expected number of computations. The analysis performed to date has been via Monte Carlo simulation and is presented in the following section.

1..5 Scale Sequential Stack Algorithm

This section details the performance of the scale sequential stack algorithm. The performance has been evaluated using a synthetic signature database and received signals formed by embedding an arbitrary signature from the database in AWGN. The performance of the stack algorithm has been computed via Monte Carlo simulation and is compared with two other detection schemes. The comparisons are based on error rate performance, computational complexity, and computational savings.

The first algorithm that is used for comparison is optimum M-ary detection. This algorithm is optimal with respect to error rate performance. A decision is formed by evaluating (9) at the finest scale (i.e. using all signal samples) for all hypotheses and choosing the hypothesis associated with the smallest metric. Using (17), the computational complexity of the algorithm can be expressed as

$$T_{opt} = K[3 \cdot 2^{J} - 1] \text{ FLOPs} \tag{19}$$

where K and J are defined as before. The second algorithm that is used for comparison employs the wavelet decomposition as described in this paper with a different decision criterion. The decision criterion is a pruning algorithm which is summarized by

$$\begin{aligned} &\text{Reject } H_k \text{ if } L^j(k) < L^j_{median} \\ &\text{Accept } H_k \text{ if } L^j(k) \geq L^j_{median} \end{aligned}$$

where $L^j(k)$ is the log-likelihood at scale j and L^j_{median} is the median log-likelihood at scale j. All hypotheses whose log-likelihoods fall below the median at a given scale are removed from consideration at finer scales. Using this decision criterion and (17), the computational complexity of the pruning algorithm is given by

$$T_{prune} = K[3 \cdot 2^{J_0} + 3(J - J_0)2^{J_0 - 1} - 1] \text{ FLOPs.} \tag{20}$$

1..6 Performance with Synthetic Database

The error rate performance has been evaluated for the stack algorithm (both the $L2$ and $L2_{norm}$ metrics) and the two algorithms detailed above. The signature database was formed from random variables uniformly distributed over the interval $[0, 1]$. The received signal was formed by adding AWGN to one of the signatures in the database. For the data to follow, a library of 200 hypotheses was used with a signal length equal to 256 samples. The performance has been computed via Monte Carlo simulation where the seed of the random number generator was varied to insure no correlation during the creation of the database and between Monte Carlo iterations. The error rate curves are presented in Figure 2. As expected, the stack algorithm with the $L2$ metric and the optimal M-ary test have an identical error rate performance. The stack algorithm with the $L2_{norm}$ metric sacrifices some error rate performance because of its sub-optimality. Additionally, Figure 2 shows that the stack algorithm with either metric is superior in error rate performance to the pruning algorithm.

The average number of FLOPs/decision is defined as the average number of FLOPs an algorithm requires to choose a hypothesis. This quantity has been obtained using the same Monte Carlo simulations used to compute the error rates. Consequently, for the data to follow the simulation parameters are unchanged. Figure 3 displays the average FLOPs/decision for the three algorithms under consideration. Once again, both the $L2$ and $L2_{norm}$ metrics are considered for the stack algorithm. The stack algorithm with the $L2$ metric offers a computational savings over the optimal maximum likelihood test for signal-to-noise ratios (SNRs) above -5dB. These savings become quite large above 5dB and the maximum savings occur above 20dB where the algorithm converges to its lower computational bound (Eq.(18)) and outperforms the pruning algorithm. The stack algorithm with the $L2_{norm}$ metric offers dramatic computational savings over the optimal M-ary test for the entire range of SNRs considered. As expected the savings are greater than those of the $L2$ metric and the algorithm converges to the lower computational bound faster (at approx. 10 dB).

The maximum computational savings offered by the stack algorithm can be best visualized be plotting the quantity

$$R = \frac{\text{FLOPs/decision for algorithm under consideration}}{\text{FLOPs/decision for optimal M-ary test.}} \tag{21}$$

This quantity is easily computed for the both the stack and pruning algorithms using (18), (19), (20) and will be denoted by R_{stack} and R_{prune} respectively. For the stack algorithm, the lower computational bound derived earlier is used for R_{stack}. It should be noted that when evaluating (21), R_{stack} is dependent on both K and J while R_{prune} is dependent on J only (J_0 is assumed fixed). Figures 4 and 5 display both R_{stack} and R_{prune} for a particular choice of parameters. In Figure 4, the number of hypothesis is held constant ($K = 10^3$) while J is varied. The stack algorithm is seen to offer greater computational savings than the pruning algorithm for all J tested. In Figure 5, the observation length is held constant ($J = 8$) while K is varied. The stack algorithm is seen to outperform the pruning algorithm for hypothesis sets with sizes in the range $[10^2, 10^6]$ which is a practical range for implementation. A more complete comparison between R_{stack} and R_{prune} is given in tables 1 and 2. The data has been rounded to facilitate comparison. From tables 1 and 2, the stack algorithm is seen to always outperform the optimal test (i.e. $R_{stack} \geq 1$). Additionally, the stack algorithm outperforms the pruning algorithm (i.e. $R_{stack} \geq R_{prune}$) for a practical range of test parameters.

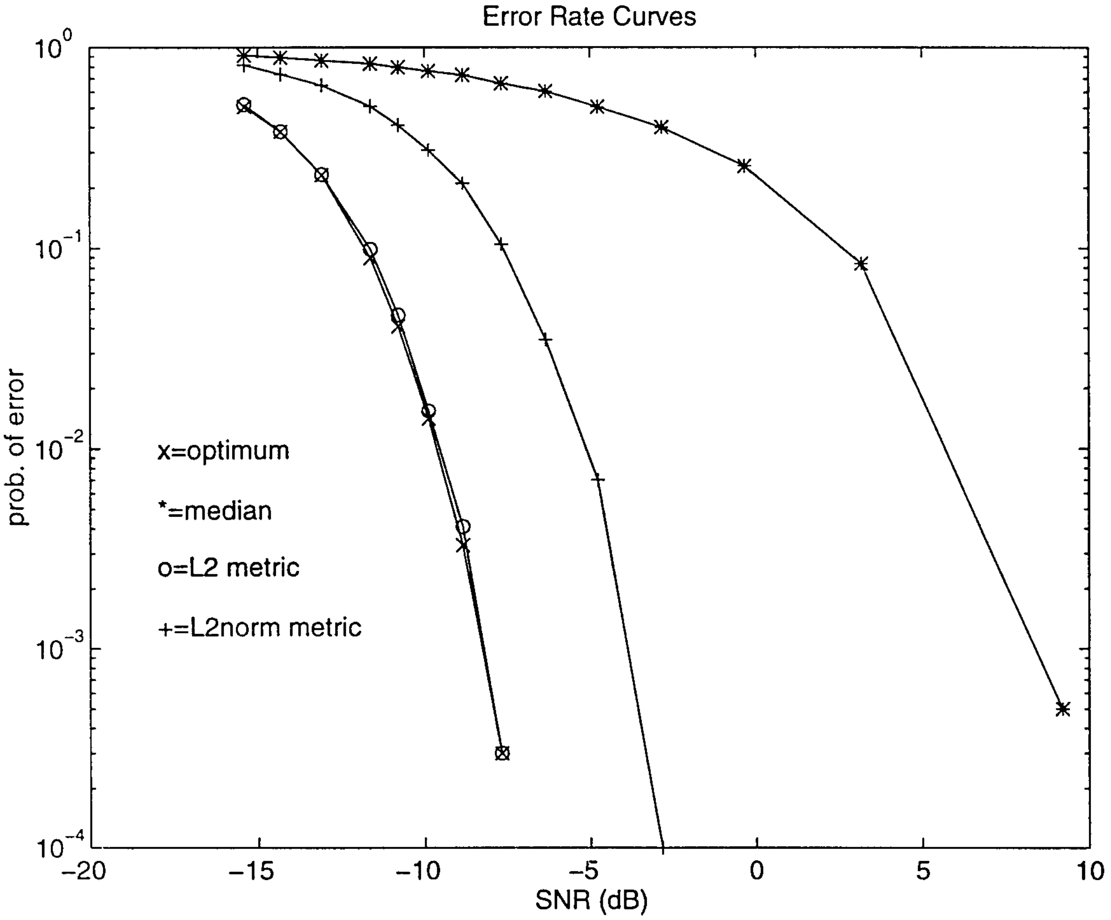

Figure 2: Error rate curves for the algorithms under consideration

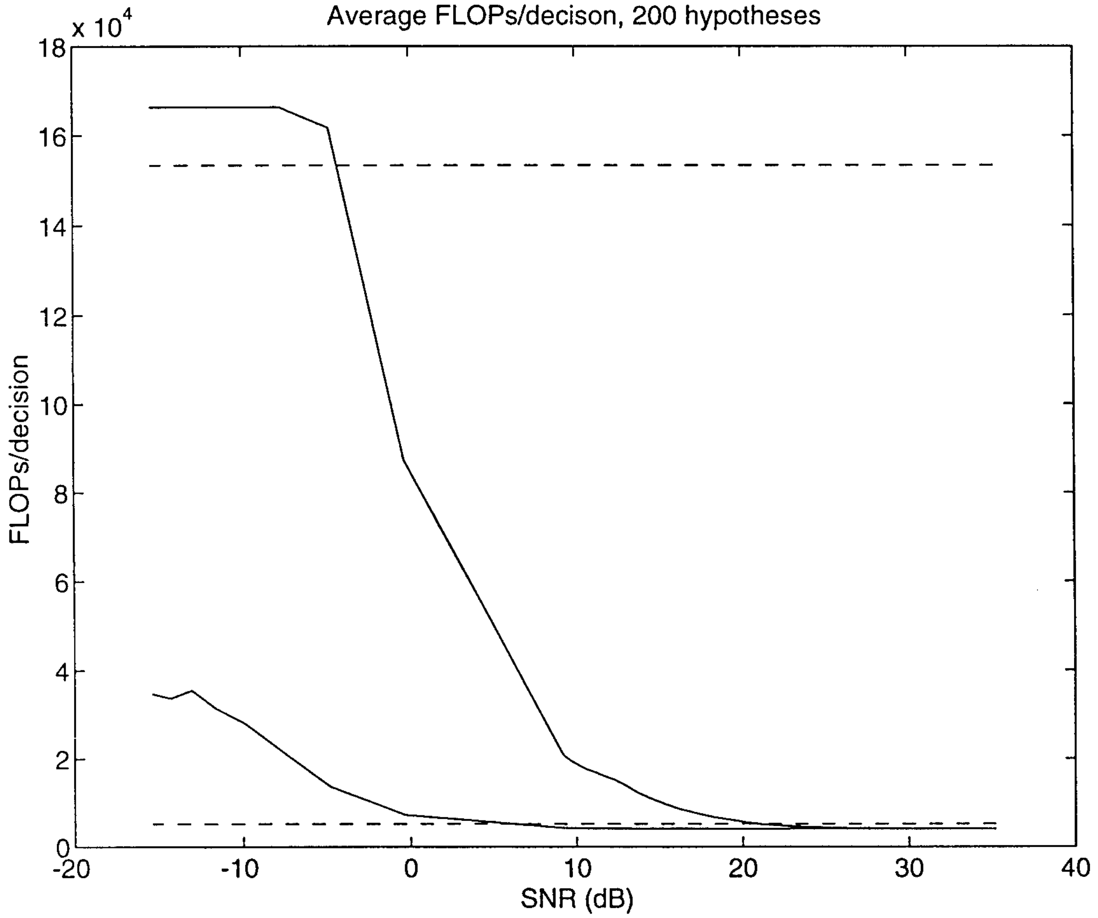

Figure 3: FLOPs/decision for the algorithms under consideration

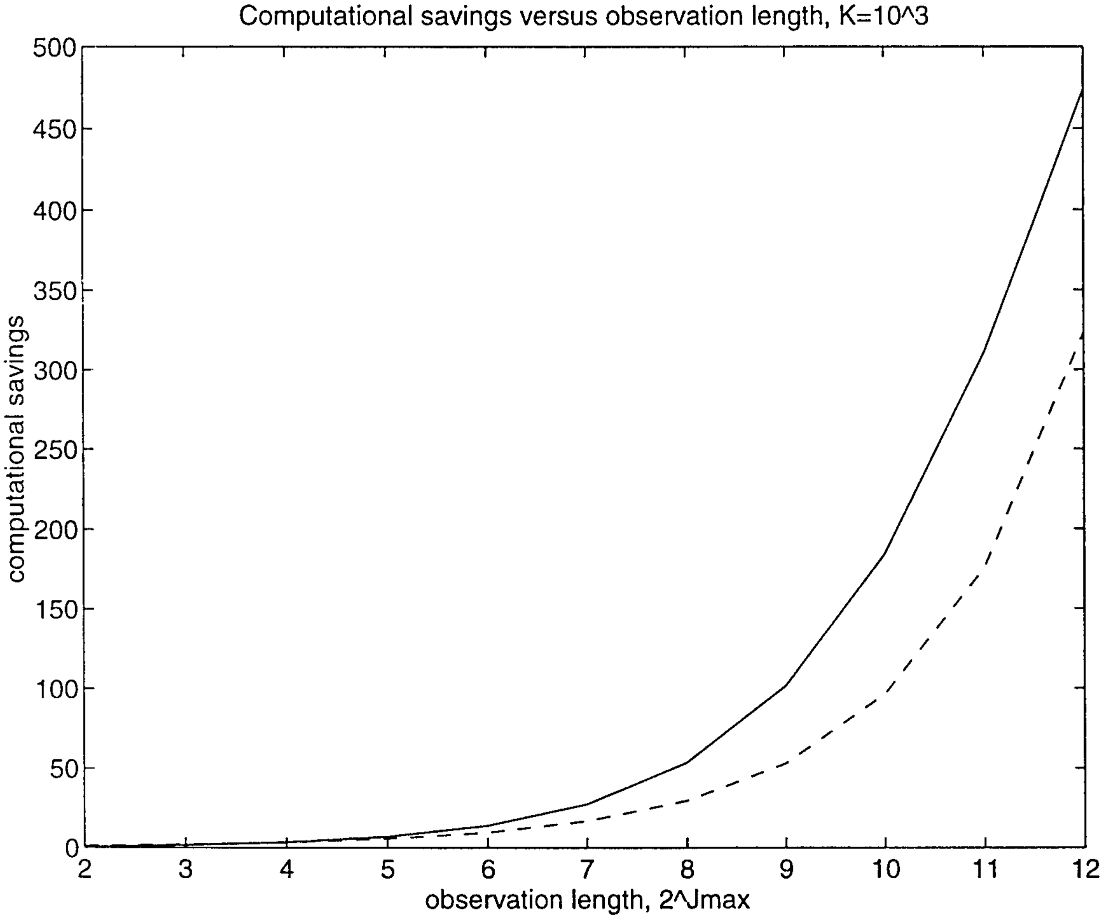

Figure 4: Computational savings versus J

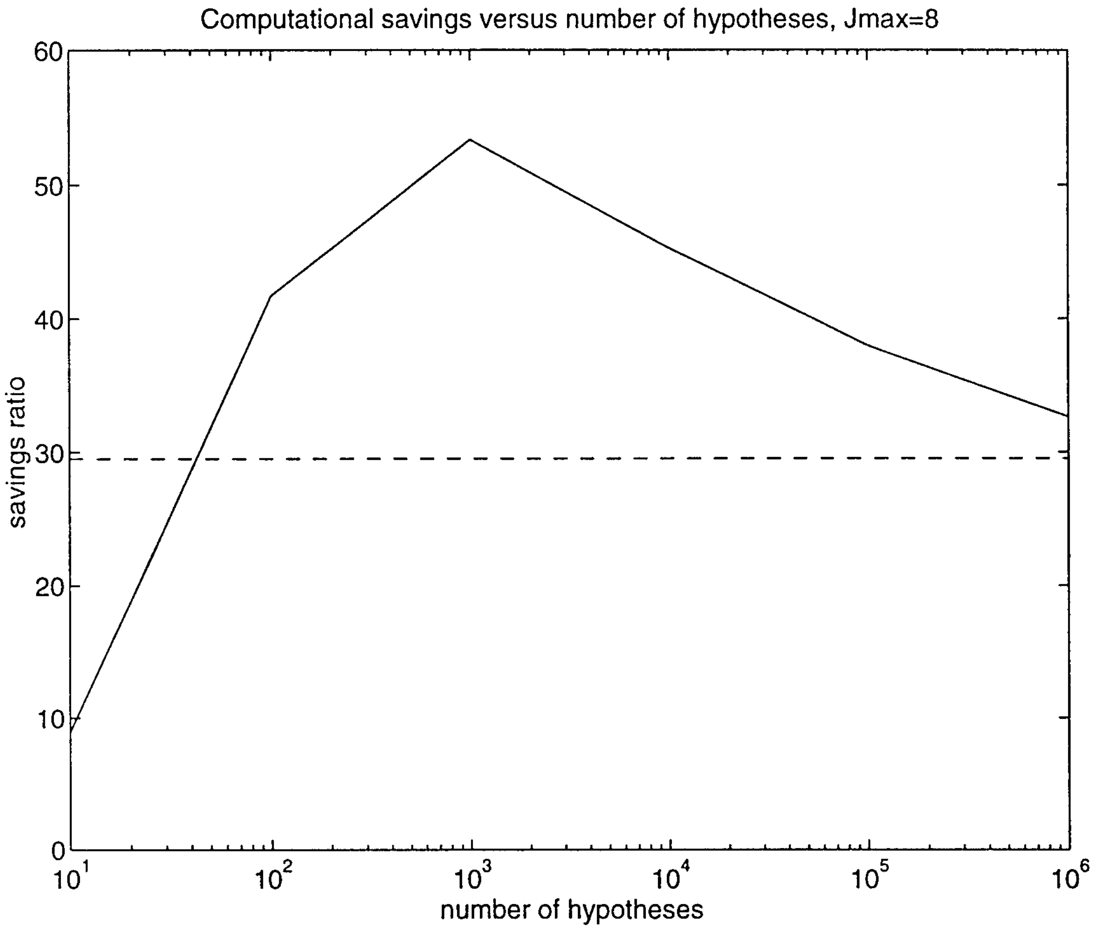

Figure 5: Computational savings versus K

Table 1: R_{stack} as a function of K and J

J	$K = 2^1$	$K = 2^2$	$K = 2^3$	$K = 2^4$	$K = 2^5$	$K = 2^6$
2	1	1	1	1	1	1
3	2	2	2	1	1	1
4	4	4	3	3	2	2
5	5	8	7	6	5	4
6	7	15	14	11	9	8
7	8	26	27	23	19	16
8	9	42	53	45	38	33
9	9	59	101	90	76	65
10	10	74	184	179	152	131
11	10	85	311	352	304	261
12	10	92	474	679	606	523

Table 2: R_{prune} as a function of K and J

J	$K = 2^1$	$K = 2^2$	$K = 2^3$	$K = 2^4$	$K = 2^5$	$K = 2^6$
2	1	1	1	1	1	1
3	2	2	2	2	2	2
4	3	3	3	3	3	3
5	6	6	6	6	6	6
6	10	10	10	10	10	10
7	17	17	17	17	17	17
8	30	30	30	30	30	30
9	53	53	53	53	53	53
10	96	96	96	96	96	96
11	176	176	176	176	176	176
12	323	323	323	323	323	323

2. A Multi-resolution based target detection algorithm – R. S. Raghavan

A multi-resolution based automatic target detection/recognition algorithm decribed in our earlier reports has been generalized to cases where a target to be detected or recognized is described only in terms of the various scales that make up the target signature. Knowledge of the actual target signature within each scale is not required in our formulation. The rationale behind this approach is that target characteristics formulated in this fashion are somewhat invariant to shifts in position and/or orientation of the target and therefore the approach provides some degree of robustness in ensuring that targets are not missed due to mismatch between the actual target signature and that used in the ATR algorithm. In our approach, target detection in clutter is carried out using a generalized likelihood ratio test that detects a signal contained in a known subspace (the signal subspace is specified completely by the scale information) embedded in clutter whose covariance properties within each scale and across scales is unknown. The clutter covariances within each scale and across scales are estimated using clutter data from target-free regions. Analytical expressions for characterizing the detection and false alarm probabilities of the generalized test have been derived. These results are further generalized to the case where the target signature is modeled as a random vector but confined to the known subspace. The proposed scale-subspace ATR algorithm may be simplified further if prior knowledge allows one to neglect clutter covariances across scale. In this case, the proposed approach has several features that make it attractive for the ATD/R application: (i) Test for target presence in each scale associated with the signal subspace is carried out independently, (ii) the false alarm probability for each scale may be specified independently, (iii) the approach offers a greater degree of protection against missing a target since the exact signature of the target is not required in the algorithm and (iv) a sequential implementation of the algorithm may be implemented readily by starting with the hypothesis test at a scale in the signal subspace with the lowest estimated clutter power and proceeding with the test towards scales with increasing estimated clutter power levels.

2..1 A Summary of Results from Previous Work

The following section summarizes the main results reported in earlier reports for the sake of establishing continuity with the work reported in the next section. Our notations are briefly introduced. These notations were introduced in an earlier report [1] and is given here so that the following document is complete. Let $x^{(J)}$ represent a complex N-dimensional vector obtained from a given resolution cell under test. The superscript J indicates the resolution level of the received data (larger the value of index J, the higher the resolution). We assume that N-vectors from a set of K secondary resolution cells or reference cells which have the same correlation properties as the test cell vector under the clutter only hypothesis are also received. These vectors from the i^{th} reference cell for example are denoted by $x^{(J)}(i)$. Using the pyrimid algorithm (figure shown in previous report) all the received vectors are decomposed into their various multi-resolution components. Thus the vector $x^{(J)}$ is equivalently represented in terms of the components $x^{(0)}, d^{(0)}, d^{(1)}, ..., d^{(J-1)}$, where $d^{(m)}$ is the detail signal at resolution level m. The dimensions of the various vectors obtained after the MRA are generally different. Since $x^{(J)}$ is a N-dimensional vector, the dimension of vector

$d^{(J-1)}$ is $N/2$, the vector $d^{(J-2)}$ has the dimension $N/4$ and so on. The last vectors in the MRA (i.e. $x^{(0)}$ and $d^{(0)}$) each have the dimension $N/2^J$. Obviously, we have assumed that $N \geq 2^J$ without loss of generality. It is convenient to stack all the vectors obtained above in a single N-vector z. Thus $z = [x^{(0)\dagger}, d^{(0)\dagger}, d^{(1)\dagger}, ..., d^{(J-2)\dagger}, d^{(J-1)\dagger}]^\dagger$

2..1.1 Adaptive detection in vector subspaces

Referring to the notations introduced in the previous section, let $z(i)$; $i = 1, 2, ..K$, $(K \geq N)$ represent a set of secondary N-vectors which are assumed to be mutually independent and to have the same statistical properties as the clutter in the primary N-vector z. Clutter in the primary vector is modeled as a zero-mean complex Gaussian N-vector with correlation matrix R_c. The condition $K \geq N$ specified above is required in order to ensure that the estimated correlation matrix (with no specific structure imposed) is non-singular with probability 1.

Let V denote the vector space spanned by the set of all complex N-vectors. Let V_1 denote the signal subspace (the vector subspace which is known to contain the signal if it is present). The dimension of the signal subspace is denoted by N_s and clearly $1 \leq N_s \leq N$. We define the subspace V_2 of dimension $N - N_s$ to be the orthogonal complement of V_1 in V. Thus, $V_1 \bigcap V_2 = 0$ and $V_1 \bigcup V_2 = V$. Since the signal subspace is known (as described in the previous section, the signal subspace is the span of vectors used as basis for the relavent scales in which the signal energy is primarily confined.) ,we assume that the components of all N-vectors have been arranged such that the projection of a given N-vector z onto to the subspace V_1 is $[z_1^t, 0_{N-N_s}^t]^t$. The superscript t denotes transpose, z_1 is a vector with N_s components and 0_M is the zero-vector of length M. Thus the projection of a given N-vector z onto the subspace V_2 is given by $[0_{N_s}^t, z_2^t]^t$, where z_2 is a vector with $N - N_s$ components. Therefore every N-vector z in the problem may be represented as $z = [z_1^t, z_2^t]^t$. We show in [1] that the generalized likelihood ratio test (GLRT) for detecting a deterministic signal vector $z_s = [z_{s1}^t, 0_{N-N_s}^t]^t$ known to belong to the subspace V_1 embedded in zero-mean complex Gaussian noise is given by:

$$\frac{z_{1.2}^\dagger S_{1.2}^{-1} z_{1.2}}{1 + z_2^\dagger S_{22}^{-1} z_2} \gtrless (y_0 - 1), \tag{22}$$

where the quantity $(y_0 - 1)$ is a predetermined threshold which as discussed below is obtained from the specified probability of false alarm. In (22) the N_s dimensional vector $z_{1.2}$ is defined as:

$$z_{1.2} = [z_1 - S_{12} S_{22}^{-1} z_2]. \tag{23}$$

Also in (22) the superscript $\dagger$ denotes conjugate transpose. Other quantities which appear in (22) and (23) are defined in terms of the secondary vector components $z_1(i)$ and $z_2(i)$; $i = 1, 2, ...K$ as follows:

$$S_{11} \;=\; \sum_{i=1}^{K} z_1(i) z_1^\dagger(i)$$

$$\begin{aligned} S_{12} &= \sum_{i=1}^{K} z_1(i) z_2^{\dagger}(i) \\ S_{21} &= \sum_{i=1}^{K} z_2(i) z_1^{\dagger}(i) \\ S_{22} &= \sum_{i=1}^{K} z_2(i) z_2^{\dagger}(i) \\ S_{1.2} &= S_{11} - S_{12} S_{22}^{-1} S_{21}. \end{aligned} \tag{24}$$

The main aspects of the test may be summarized as follows. The secondary vectors provide estimates of the clutter correlation and cross-correlation matrices in the two subspaces V_1 and V_2 as shown in equation (24). These estimated matrices are needed for estimating the component of the primary clutter vector in the signal subspace V_1 using the component of the primary vector in subspace V_2 which is known to be signal-free. The estimated clutter vector is given by $\hat{z}_1 = S_{12} S_{22}^{-1} z_2$. Therefore, $z_{1.2} = z_1 - \hat{z}_1$ represents a N_s dimensional vector whose clutter level has been suppressed through linear estimation and which in addition may contain the signal to be detected. Note that the signal to be detected (if present) remains unaltered through this processing and the clutter suppression attempts to increase the signal-to-clutter ratio in the subspace V_1. Next, the residual clutter vector in the signal subspace V_1 is 'whitened' using the estimated correlation matrix of the residual clutter in V_1. This matrix may be shown to be proportional to $S_{1.2}$ defined in (24). The vector $S_{1.2}^{-1/2} z_{1.2}$ represents the output of the whitening filter and the numerator of the left hand side of (22) is the energy of the vector output of the whitening filter. The GLRT compares this quantity (a real and non-negative scalar) to the quantity $(y_0 - 1)(1 + z_2^{\dagger} S_{22}^{-1} z_2)$, which represents an adaptive threshold determined from the primary and secondary clutter components in subspace V_2. Note that the equivalent of the Adaptive Matched Filter (AMF) described in [5] for the present problem is to compare the numerator of (22) with a fixed threshold instead of the adaptive threshold used to implement the GLRT. A block diagram summarizing these ideas is shown in Fig. 6

2..2 Summary of New Results

The new work reported here include the following:

Analytical expressions (in closed form) for the probability of detection as a function of the signal-to-clutter ratio for the subspace algorithm.

Generalization of the results for the case where the target signature is a random vector known to be contained in a given subspace.

2..2.1 Probability of Detection For Subspace Algorithm

In this section we obtain analytical expressions for the probability of false alarm P_{FA}, and the probability of detection P_D for the GLRT in (22). We define the following quantities for convenience:

$$y = z^{\dagger} S^{-1} z$$

$$\begin{aligned} y_{22} &= z_2^\dagger S_{22}^{-1} z_2 \\ y_{1.2} &= z_{1.2}^\dagger S_{1.2}^{-1} z_{1.2}. \end{aligned} \tag{25}$$

We first obtain the expression for P_D for the GLRT in (22) as a function of the signal-to-interference ratio defined as: $x = z_s^\dagger R_i^{-1} z_s$, where z_s is the signal to be detected. The probability of false alarm is obtained by setting the signal-to-interference ratio to zero in the expression for P_D. Using upper case symbols to denote random variables, we observe that $Y = Y_{22} + Y_{1.2}$ under both hypotheses H_0 and H_1. Let the random variable R denote $1/(1+Y_{22})$ and so $Y = Y_{1.2} + R^{-1} - 1$. Since $Y_{1.2}$ is a non-negative random variable, the conditional probability of $Y > (y_0 - 1)$ given $R = r$ is 1 if $0 < r < 1/y_0$. For $1/y_0 < r < 1$, the above conditional probability is $P[RY_{1.2} > (ry_0 - 1)|R = r, 1/y_0 < r < 1]$. The probability of $Y > (y_0 - 1)$ may be computed by removing the conditioning for the random variable R. Thus:

$$\begin{aligned} P[Y > (y_0 - 1)|H_1, x] &= \int_0^{1/y_0} f_R(r)dr \\ &+ \int_{1/y_0}^1 f_R(r) P[RY_{1.2} > (ry_0 - 1)|H_1, R = r, x]dr. \end{aligned} \tag{26}$$

The probability on the left hand side was derived in [2] and is given by:

$$P[Y > (y_0 - 1)\ |H_1, X = x] = 1 - \frac{1}{y_0^K} \sum_{i=1}^{K-N+1} \binom{K}{N+i-1} (y_0 - 1)^{N+i-1} G_i(x/y_0), \tag{27}$$

where, $G_i(x)$ is related to the incomplete Gamma function $\Gamma(i, x)$ by:

$$G_i(x) = e^{-x} \sum_{n=0}^{i-1} \frac{x^n}{n!} = \frac{\Gamma(i, x)}{(i-1)!}. \tag{28}$$

Also, the random variable R is Beta distributed under both hypotheses (since primary vector components from only the subspace V_2, which is signal-free, are involved). Thus,

$$f_R(r) = \frac{K!}{(K - N + N_s)!(N - N_s - 1)!} r^{K-N+N_s} (1 - r)^{N-N_s-1}, \qquad 0 \le r \le 1. \tag{29}$$

We seek to develop an expression similar to that in [3] for the probability of $RY_{1.2} > (y_0 - 1)$ conditioned on hypothesis H_1, signal-to-interference ratio x and $R = r$. By analogy with expressions in [3] and [4] we assume that the required solution is given in the form shown below but with the terms $C_i G_i(xr/y_0)$ appearing in the sum on the right hand side. The unknown coefficients C_i are then determined by substituting this expression in (26) and using (27,28,29) (details of the integrations involved are described in [4]). Thus:

$$\begin{aligned} P[RY_{1.2} > (y_0 - 1)|H_1, x, R = r] &= \\ 1 - \frac{1}{y_0^{K-N+N_s}} \sum_{i=1}^{K-N+1} \binom{K - N + N_s}{N_s + i - 1} (y_0 - 1)^{N_s+i-1} G_i(xr/y_0). \end{aligned} \tag{30}$$

The probability of detection P_D is obtained by averaging the conditional probability of detection in (30) over the probability density function of the random variable R, given in (29). For this purpose, we first define the function $D_i(.)$ as:

$$D_i(x) = \int_0^1 G_i(xr) f_R(r) dr. \tag{31}$$

The probability of detection for the GLRT is given by:

$$P_D = 1 - \frac{1}{y_0^{K-N+N_s}} \sum_{i=1}^{K-N+1} \binom{K-N+N_s}{N_s+i-1} (y_0 - 1)^{N_s+i-1} D_i(x/y_0), \tag{32}$$

The probability of false alarm P_{FA} is obtained by setting the signal-to-interference ratio to zero in the above expression ($D_i(0) = 1$) and is:

$$P_{FA} = 1 - \frac{1}{y_0^{K-N+N_s}} \sum_{i=1}^{K-N+1} \binom{K-N+N_s}{N_s+i-1} (y_0 - 1)^{N_s+i-1} \tag{33}$$

An alternate expression for the probability of false alarm which is convenient for computation is obtained by expanding the term within the sum above in a binomial series and is given by:

$$P_{FA} = \binom{\bar{K}}{\bar{N}-1} \left(\frac{1}{y_0}\right)^{\bar{K}-\bar{N}+1} {}_2F_1(\bar{K}-\bar{N}+1, 1-\bar{N}; \bar{K}-\bar{N}+2; y_0^{-1}), \tag{34}$$

where $\bar{K} = K - (N - N_s)$, $\bar{N} = N - (N - N_s) = N_s$ and ${}_2F_1(\alpha, \beta; \gamma; y)$ is the Gauss Hypergeometric series given by:

$$ {}_2F_1(\alpha, \beta; \gamma; y) = 1 + \sum_{k=1}^{\infty} \frac{(\alpha)_k (\beta)_k}{(\gamma)_k} \frac{y^k}{k!}, \tag{35}$$

where,

$$(\theta)_k = \prod_{i=0}^{k-1} (\theta + i) \qquad ; k \geq 1. \tag{36}$$

Note that when $N_s = 1$, the P_{FA} evaluated from (34) is $1/y_0^{K-N+1}$ which coincides with the expression in [3]. We need to define a few quantities before the above expressions may be applied to the case when $N_s = N$. When $N_s = N$, the subspace V_2 has dimension zero. Accordingly, we define $Y_{1.2} = Y$, $Y_{22} = 0$ and so $f_R(r) = \delta(r-1)$. Thus, $D_i(x)$ in (31) is equal to $G_i(x)$ and as a result the P_{FA} evaluated from (34) coincides with the expression in [4]. We also note that the expressions for P_D given in (32) for $N_s = 1$ and $N_s = N$ coincide with the expressions in [3] and [4] respectively.

2..2.2 Probability of Detection of the Subspace Algorithm for Random Signals

The probability of detecting a deterministic signal z_s which is known to belong to the signal subspace V_1 using the GLRT in (22) as derived in the previous section is given in terms of the signal-to-interference ratio $x = z_s^{\dagger} R_i^{-1} z_s$ as:

$$P_D = 1 - \frac{1}{y_0^{\bar{K}}} \sum_{i=1}^{\bar{K}-\bar{N}+1} \binom{\bar{K}}{\bar{N}+i-1} (y_0 - 1)^{\bar{N}+i-1} D_i(x/y_0), \tag{37}$$

where $D_i(.)$ is a function defined in (31). The above expressions for P_{FA} and P_D are valid for any dimension of the signal subspace V_1 between one and N (i.e. $1 \leq N_s \leq N$).

When the signal to be detected is a realization of a random vector Z_s confined to the subspace V_1, (22) is not the GLRT for the detection problem. In fact if the covariance matrix of the signal and interference are both unspecified the GLRT is intractable [7]. To quantify the detection performance of the test in (22) for the random signal case, we assume as before that the signal vector Z_s is expressed in the form $[Z_{s1}^t \; 0_{N-N_s}^t]^t$. The vector Z_{s1} is modeled as a complex, zero-mean Gaussian random vector of dimension N_s with covariance matrix R_{s11}. The probability of detection for the random signal case is obtained by averaging the probability in (37) which is conditioned on a given signal-to-interference ratio $X = x$ over the probability density function of $X = Z_s^\dagger R_i^{-1} Z_s$. Given $Z_s = [Z_{s1}^t \; 0_{N-N_s}^t]^t$, the random variable $X = Z_s^\dagger R_i^{-1} Z_s$ is also expressed in terms of the partioned interference covariance matrix as : $X = Z_{s1}^\dagger R_{i1.2}^{-1} Z_{s1}$, where $R_{i1.2} = R_{i11} - R_{i12} R_{i22}^{-1} R_{i21}$. In a manner similar to that described in [4] it may be shown that the probability density function of the random variable X is determined by the eigenvalues of the matrix: $R_{s11} R_{i1.2}^{-1}$. Specifically, if $\{\lambda_i \; ; i = 1, 2, ... N_s\}$, denote the eigenvalues of the matrix $R_{s11} R_{i1.2}^{-1}$, then the random variable X is statistically equivalent to the sum: $\sum_{i=1}^{N_s} \lambda_i X_i$, where X_i are statistically independent exponentially distributed random variables with the mean value of each random variable equal to 1. Thus,

$$\begin{aligned} P_D &= \int_0^\infty P[Y_{1.2}/(1+Y_{22}) > (y_0 - 1) | X = x, H_1] f_X(x) dx \\ &= 1 - \frac{1}{y_0^{\bar{K}}} \sum_{i=1}^{\bar{K}-\bar{N}+1} \binom{\bar{K}}{\bar{N}+i-1} (y_0 - 1)^{\bar{N}+i-1} \bar{D}_i(y_0^{-1}), \end{aligned} \tag{38}$$

where,

$$\bar{D}_i(\alpha) = \int_0^\infty D_i(\alpha x) f_X(x) dx. \tag{39}$$

The average signal-to-interference ratio λ_0, is defined as : $\lambda_0 = E[Z_s^\dagger R_i^{-1} Z_s] = tr[R_s R_i^{-1}] = tr[R_{s11} R_{i1.2}^{-1}]$, where $tr[\;]$ denotes the trace operator. Since the trace of a matrix is also equal to the sum of the eigenvalues of the corresponding matrix, the average signal-to-interference ratio is also given by $\lambda_0 = \sum_{i=1}^{N_s} \lambda_i = E[X]$. Therefore, the probability of detection for given P_{FA}, K, N and $1 < N_s \leq N$ is not parameterized by the single quantity λ_0–the average signal-to-interference ratio, as in the deterministic signal case, but P_D for the random signal case depends on all the eigenvalues $\lambda_i \; ; \; i = 1, 2, ..N_s$ through the density function $f_X(x)$ that appears in (39).

2..3 Summary

In the previous section we have generalized the adaptive subspace detection algorithm (reported previously in [1]) to the case when the signal to be detected is random but confined to a signal subspace. Analytical expressions for the detection probability are derived. To the best of our knowledge these results are new to the field of adaptive detection theory. A more detailed version of these results is published in an article entitled "Performance of the GLRT for Adaptive Vector Subspace Detection," which appears in the IEEE Transactions on Aerospace and Electronic Systems, October 1996.

References

[1] Section 2 of Progress Report for ARPA grant F49620-93-1-0490, April 1995.

[2] E. J. Kelly and K. M. Forsythe, "Adaptive Detection and Parameter Estimation for Multidimensional Signal Models," Technical Report 848, Lincoln Laboratory, Massachusetts Institute of Technology, 1989.

[3] E. J. Kelly, " An Adaptive Detection Algorithm," IEEE Trans. on Aerospace and Electronic Systems, vol. AES-22, No. 1, pp. 115 -127, Mar. 1986.

[4] R. S. Raghavan, H. F. Qiu and D. J. McLaughlin, "CFAR Detection in Clutter with Unknown Correlation Properties," IEEE Trans. on Aerospace and Electronic Systems, vol. AES-31, no.2, pp. 647-657, April 1995.

[5] F. C. Robey, D. R. Fuhrmann, E. J. Kelly and R. Nitzberg, "A CFAR Adaptive Matched Filter Detector," IEEE Trans. on Aerospace and Electronic Systems, vol. AES-28, no.1, pp. 208 - 216, Jan. 1992.

[6] S. Bose and A. O. Steinhardt, "The Optimum Array Detector for a Weak Signal in Unknown Noise," IEEE Trans. on Aerospace and Electronic Systems, vol. AES-32, no.3, pp. 911-922, July 1996.

[7] S. Bose and A. O. Steinhardt, "A Maximal Invarient Framework for Adaptive Detection with Structured and Unstructured Covariance Matrices," IEEE Trans. on Signal Processing, vol. 43, no. 9, pp. 2164-2175, Sep. 1995.

[8] L. L. Scharf and B. Friedlander, "Matched Subspace Detectors," IEEE Trans. on Signal Processing, vol. 42, no. 8, pp. 2146-2157, Aug. 1994.

[9] L. L. Scharf, "Statistical Signal Processing: Detection, Estimation and Time Series Analysis," Chapter 4, Addison-Wesley Publishing Company, Inc., New York, 1991.

[10] R. J. Muirhead, "Aspects of Multivariate Statistical Theory," Chapter 6, John Wiley & Sons, Inc., New York, 1982.

[11] E. L. Lehmann, "Testing Statistical Hypotheses," Second Edition, Wadsworth, Inc., Belmont, California, 1991.

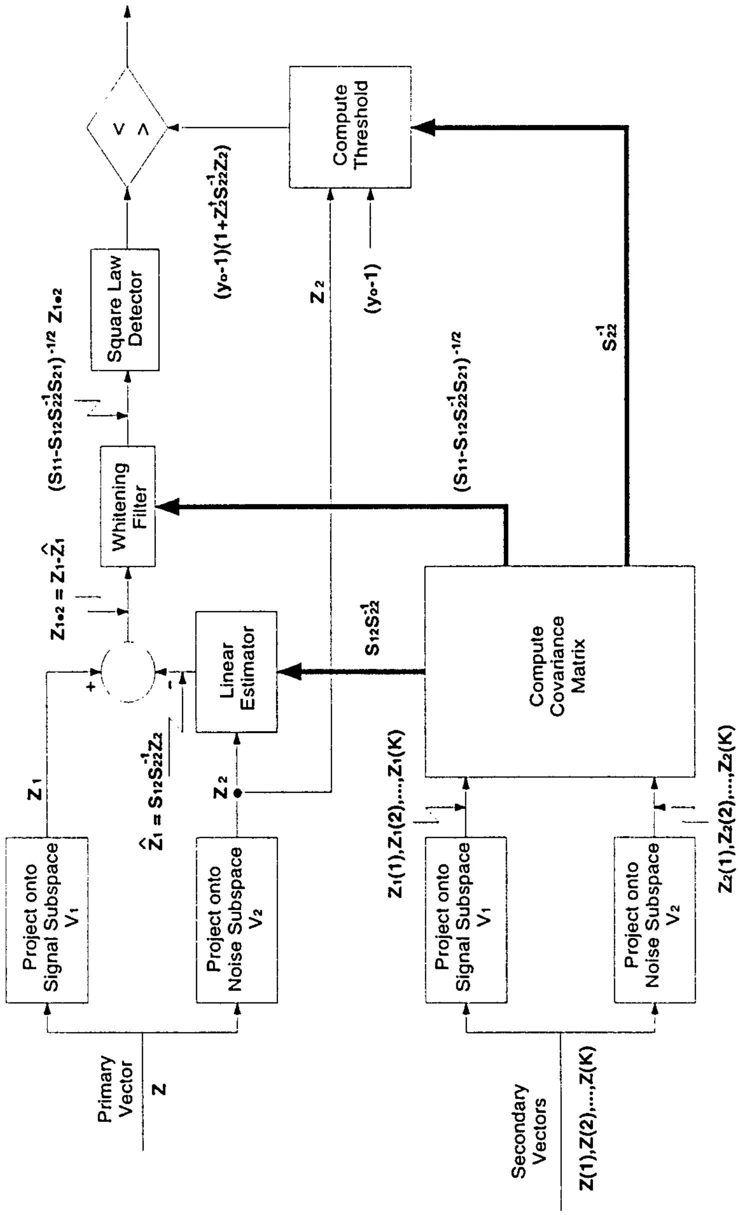
Primary Vector
Z
Secondary Vectors
Z(1),Z(2),...,Z(K)
Project onto Signal Subspace V₁
Project onto Noise Subspace V₂
Project onto Signal Subspace V₁
Project onto Noise Subspace V₂
Z_1
Z_2
$\hat{Z}_1 = S_{12}S_{22}^{-1}Z_2$
+
-
Linear Estimator
$Z_1(1),Z_1(2),\ldots,Z_1(K)$
$Z_2(1),Z_2(2),\ldots,Z_2(K)$
Compute Covariance Matrix
$S_{12}S_{22}^{-1}$
$Z_{1\bullet 2} = Z_1 - \hat{Z}_1$
Whitening Filter
$(S_{11}-S_{12}S_{22}^{-1}S_{21})^{-1/2}$
$(S_{11}-S_{12}S_{22}^{-1}S_{21})^{-1/2}Z_{1\bullet 2}$
Square Law Detector
Z_2
(y_o-1)
S_{22}^{-1}
Compute Threshold
$(y_o-1)(1+Z_2^{\dagger}S_{22}^{-1}Z_2)$
> <

Figure 6: Block Diagram of Subspace GLRT Algorithm.

3. Scale-Recursive Optimal Filtering – H. Lev-Ari

This research has been conducted by Professor Hanoch Lev-Ari and the graduate student Hong Liu. The results reported here are extracted from the reports: (i) H. Liu and H. Lev-Ari, "Scale-Recursive Optimal Filtering," CDSP Technical Report No. TR-CDSP-94-25, Northeastern University, Dec. 1994 (see also *1995 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 2076-2079, Detroit, MI, May 1995); and (ii) H. Liu and H. Lev-Ari, "Optimized Filter Banks for Efficient Subband-Domain Estimation," CDSP Technical Report No. TR-CDSP-96-38, Northeastern University, Feb. 1996. A journal article is currently in preparation.

Multiresolution analysis decomposes a single record into a hierarchy of signals at different scales, i.e., into a multichannel configuration (Fig. 7). This enables the application of efficient channel-recursive estimation techniques to construct optimal (Wiener) and adaptive filters that operate recursively from coarse to fine scale.

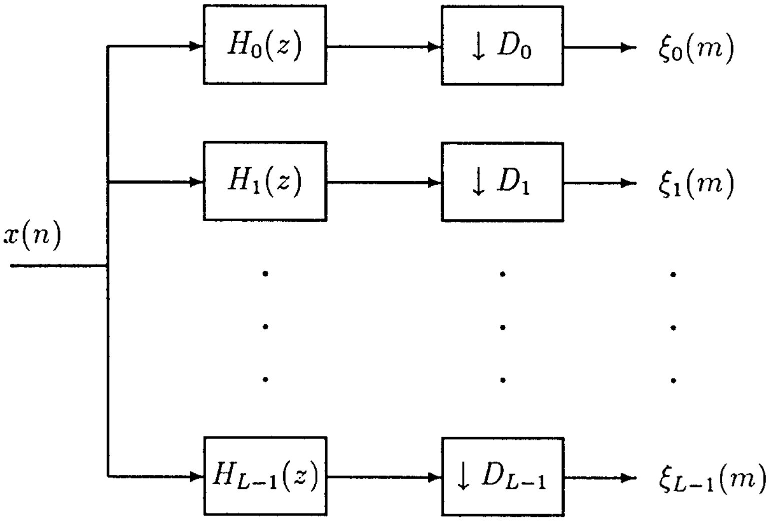

Figure 7: Subband-domain decomposition of a signal $x(\cdot)$.

We have developed [1] an analytical framework for the implementation of optimal filters in the multiresolution (i.e., subband) domain. Our approach to scale-recursive optimal filtering is based on carrying out the filtering operation in the subband domain, using a selected *subset* of the subband-domain signals. This approach allows to trade-off performance for computational cost: using a fraction of the number of subband channels we can achieve estimation errors that are close to the theoretical minimum, and at a fraction of the cost associated with using the complete set of subband channels. Though some results on matching wavelet-based filter banks to specific signals have already been published [2, 3, 4, 5], previous research has not addressed the issue of cost-performance tradeoff in subband-domain optimal estimation.

To be more specific, consider the (standard) problem of estimating a signal (say, target signature) $d(\cdot)$ from a received (noisy) signal $x(\cdot)$. This can be accomplished by the celebrated Wiener filter [6], which determines an optimal estimate $\hat{d}(\cdot)$ via linear filtering operation, viz., $\hat{d} = T \circledast x$, where $T(z)$ denotes the transfer function of the Wiener filter (Fig. 8a). In contrast, we advocate carrying out the filtering operation in the subband domain (Fig. 8b), and using a selected *subset* of the subband-domain signals $\xi_k(\cdot)$, viz., $\widehat{\boldsymbol{\delta}}^{(i)} = \mathcal{T}^{(i)} \circledast \boldsymbol{\xi}^{(i)}$, where

$$\boldsymbol{\xi}^{(i)}(m) := \begin{pmatrix} \xi_0(m) \\ \xi_1(m) \\ \vdots \\ \xi_i(m) \end{pmatrix} \tag{40}$$

Here, $\{\delta_i(m)\,;\ 0 \le i \le L-1\}$ denote the subband components of the signal $d(\cdot)$, $\boldsymbol{\delta}^{(i)}$ is a partial subset of such channels (analogous to $\boldsymbol{\xi}^{(i)}$), $\mathcal{T}^{(i)}$ denotes the response of the subband-domain filter, and $\widehat{\boldsymbol{\delta}}^{(i)}$ is an estimate of $\boldsymbol{\delta}^{(i)}$.

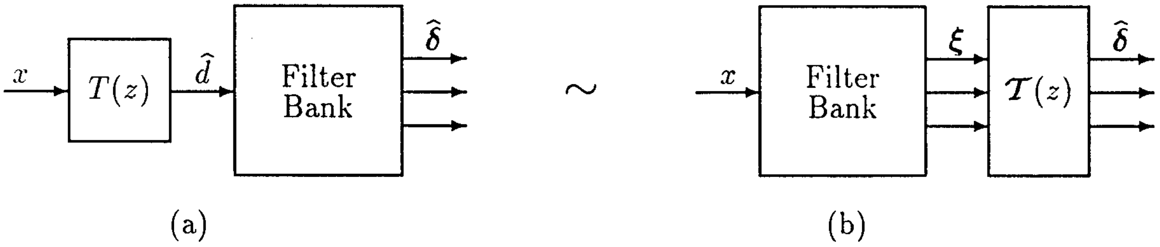

Figure 8: Linear filtering in: (a) input-domain vs. (b) subband-domain

Our approach is motivated by the consideration that $\widehat{\delta}_k(\cdot)$, the estimated subband components of the desired signal $d(\cdot)$, are to be used as an input for further processing, such as target detection and/or classification. Moreover, we assume that such further processing is to be done in the subband domain as well. This dictates our choice to produce estimates of the subband components $\delta_k(\cdot)$, rather than directly estimating $d(\cdot)$ itself.

We have quantified the trade-off between quality of performance and implementation cost associated with subband estimation schemes. We have demonstrated [1] how this trade-off can be controlled by several distinct factors, including:

- Resolution level – total number of available subband channels.
- Channel overlap – degree of overlap between the frequency responses of subband channels. Is determined by the choice of a "mother wavelet."
- Utilization ratio – the fraction of available subband channels that are involved in the estimation scheme (given by the ratio between the integer i in (40) and the integer L, the total number of available subband channels).
- Inter-channel estimation sparsity – the number of neighboring channels used in forming the estimate $\widehat{\delta}_i(\cdot)$. This index may range from a value of zero, which corresponds to the diagonal estimation scheme, through a value of 1 (tridiagonal scheme), to the full-complexity variant, which uses all subband channels.
- Filter-order, or intra-channel estimation sparsity – the number of samples from each of the subband channels used in estimation of $\delta_i(n)$ for every time instant n.

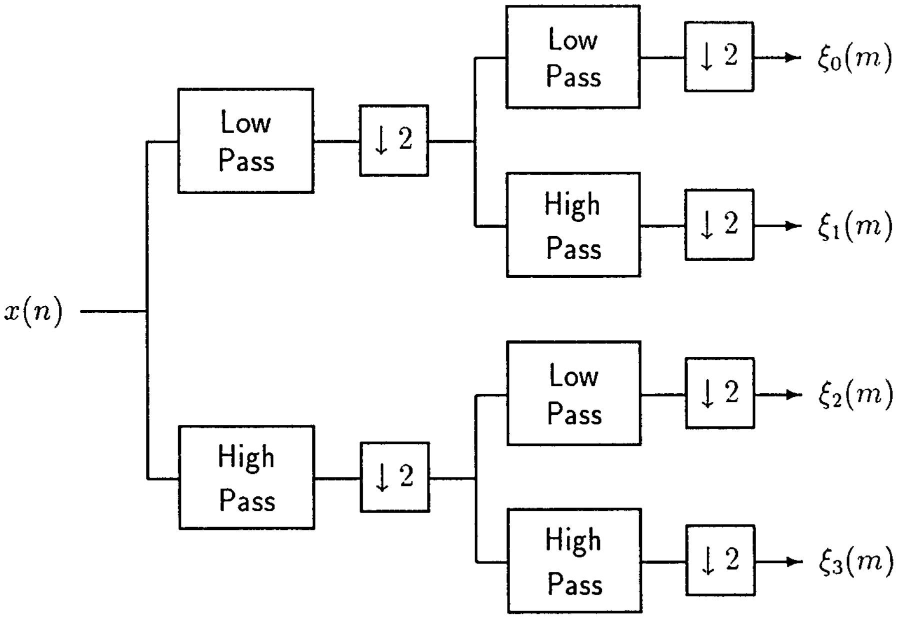

Figure 9: Wave-packet configuration of filter bank

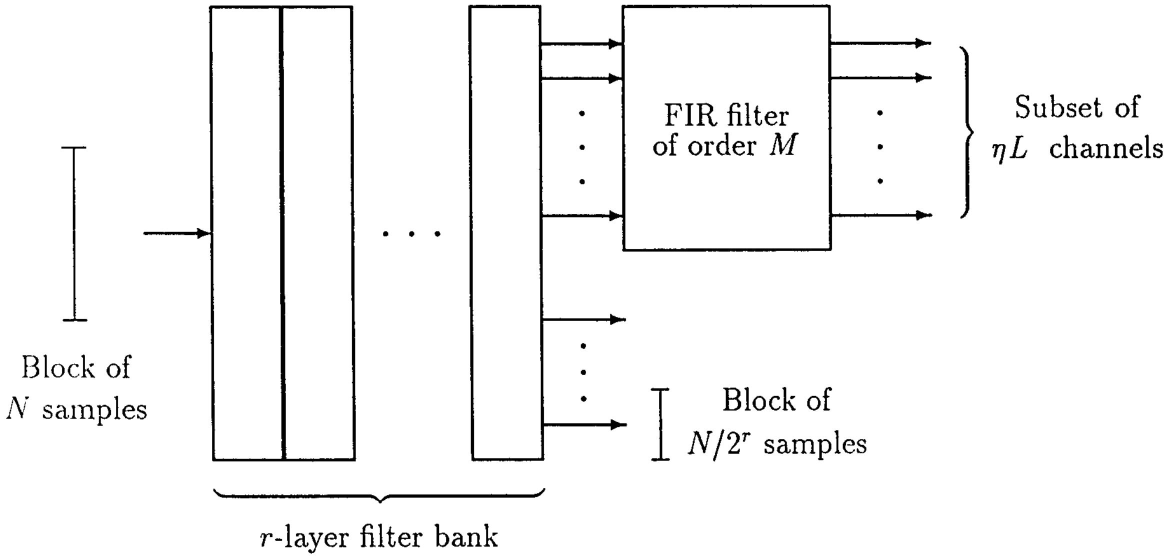

Figure 10: Structure of subband-domain (suboptimal) estimator.

The first two factors are determined by the details of the subband decomposition scheme itself. For instance, the level of resolution in wavelet- and wavepacket-based filter-bank configurations is determined by the depth of the binary tree that characterizes the filter-bank, while the channel overlap depends also on the frequency characteristics of the prototype lowpass filter: lower overlap can be achieved by higher quality filters, but at the expense of increased computational cost. We use a filter bank with a binary-tree configuration (Fig. 9), combined with a sparse subband-domain estimator (Fig. 10).

The remaining three factors depend on the details of the (optimal) estimation scheme used. The utilization ratio characterizes the subset of subband channels that are actually involved in the estimation process: ignoring (i.e., not estimating) some of the subband components $\delta_i(\cdot)$ results in a reduction in computational cost, combined with some degradation in overall performance. The notion of estimation sparsity offers a further refinement of the same cost-performance trade-off: relying on the relatively low overlap between non-adjacent subband channels we can significantly reduce the cost of computing each individual subband estimate $\delta_i(\cdot)$, which suffering only minor increase in the attendant estimation error. Finally, selecting the order of the estimation filter (including the possibility of channel-dependent order) is get another way to control the cost-performance trade-off. While order selection is the only degree of freedom available in classical optimal filtering, here it combines with the other four factors to offer a much broader range of design scenarios.

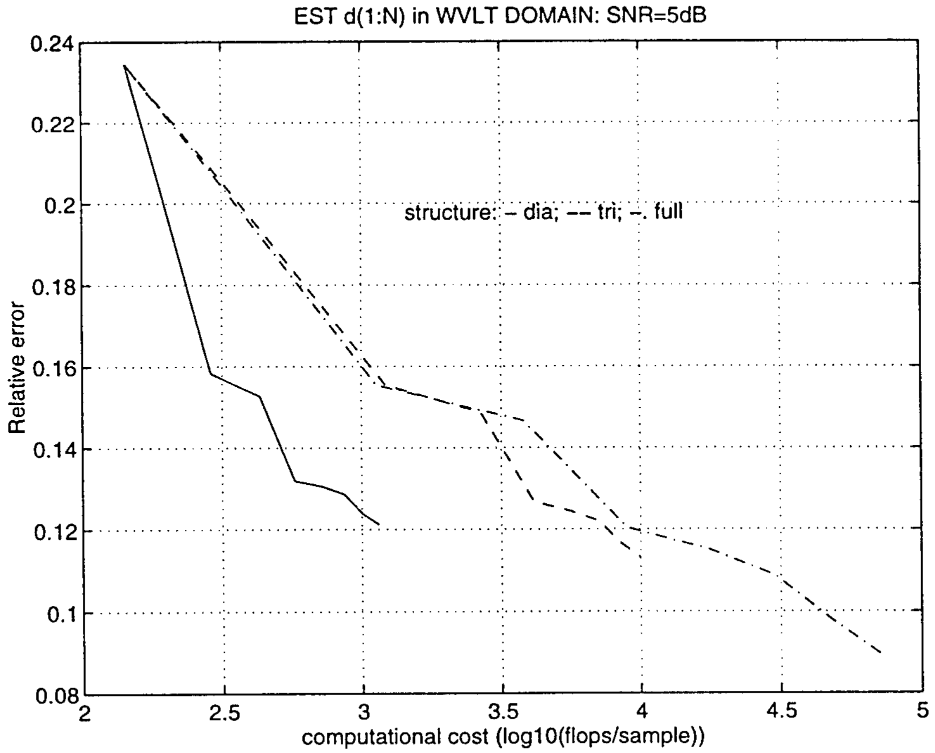

Figure 11: Cost-performance tradeoff: utilization ratio and inter-channel sparsity

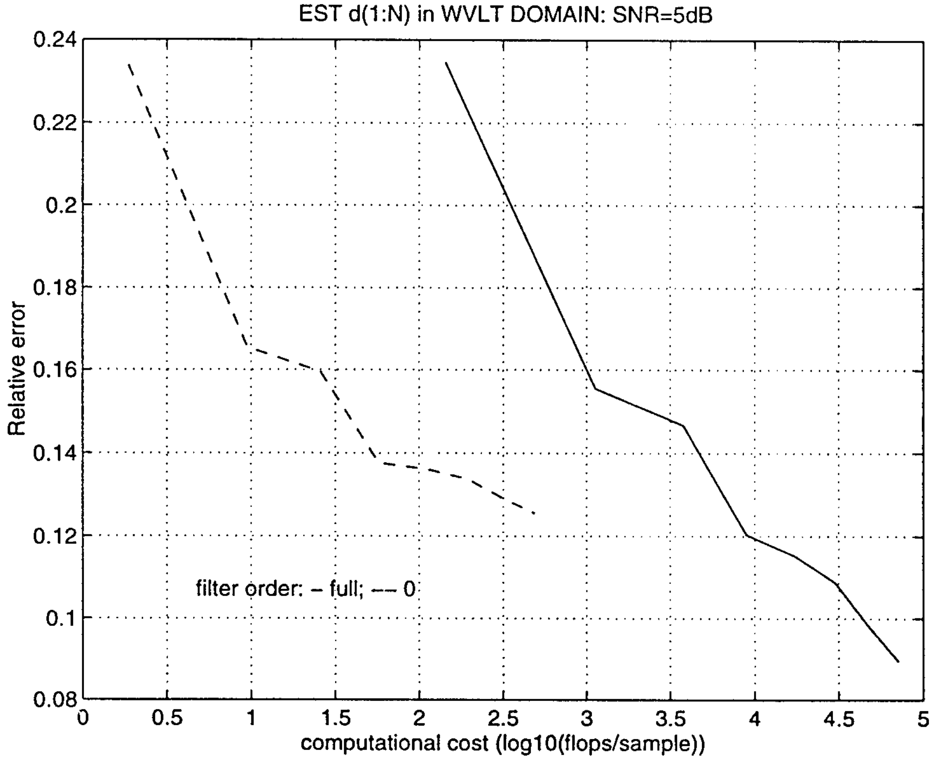

Figure 12: Cost-performance tradeoff: utilization ratio and intra-channel sparsity

In Sec. 3..1 we introduce the notions of channel overlap and estimation sparsity. We describe the structure of a proposed subband-domain estimator that incorporates the five design factors mentioned earlier. In Sec. 3..2 we describe the effect of these factors on the cost-performance trade-off. We show that utilization ratio and estimation sparsity tend to be the dominating factors in the sense that they allow the most significant reduction of computational cost for a given level of estimation error, and that the most effective cost-performance tradeoff is achieved with a memoryless, diagonal estimator configuration, i.e., with the highest level of both inter-channel and intra-channel sparsity (see, e.g., Figs. 11,12). Consequently, we have focused on optimizing the configuration of the tree-structured filter bank. This means we have compared the various possible subgraphs (actually subtrees) of the wave-packet configuration of Fig. 9 in terms of their cost-performance tradeoff and constructed procedures for optimal selection of such subtrees.

Such a comparison can be conveniently illustrated by a tradeoff chart showing estimation error (our measure of performance quality) vs. computational cost, as in Fig. 13. In this chart every point (indicated by an x) represents a particular subtree, i.e., its coordinates are the estimation error $\mathcal{E}$ and the computational cost $\mathcal{C}$ associated with this subtree. The convex boundary curve at the bottom of the chart characterizes the collection of optimal subtrees: the subtrees on this curve are optimal in the sense that they minimize the cost function

$$J = \lambda\mathcal{C} + (1-\lambda)\mathcal{E} \quad , \qquad 0 \leq \lambda \leq 1 \tag{41}$$

which is a convex combination of the estimation error $\mathcal{E}$ and the computational cost $\mathcal{C}$. A particular subtree on the boundary curve corresponds to a particular choice of the weight parameter λ: as λ varies from 0 to 1, we obtain every point on the boundary curve, from bottom right to top left.

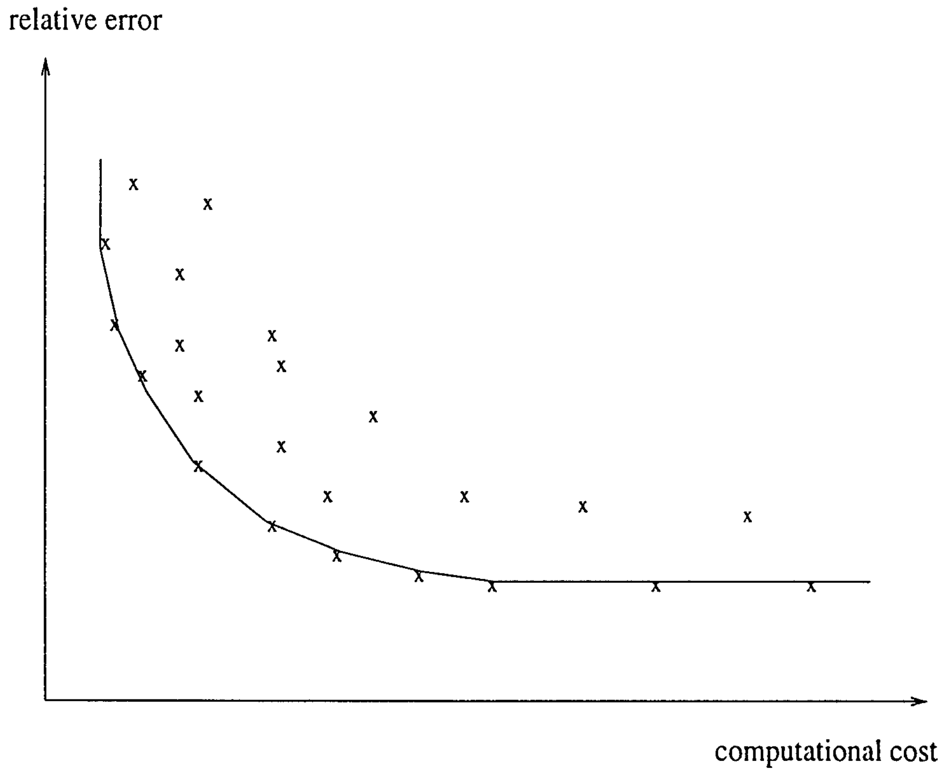

Figure 13: The cost-performance chart: comparison of subtrees

In Sec. 3..3 we present an algorithm for obtaining the optimal filter-bank configuration for every specific value of the weight parameter λ. The collection of such configurations obtained as λ ranges from 0 to 1 constitutes the *optimal family*, i.e., the subtrees that determine the boundary curve discussed above. Since the optimal family contains a *finite* number of subtrees, the boundary curve is piecewise-linear. This means that the range of values of the weight parameter λ is divided into subranges, where each subrange corresponds to a single subtree from the optimal family. Since these subranges are not known apriori, the construction of the optimal family (as well as of the associated boundary curve) requires a search over the entire range of λ.

As an alternative, we present in Sec. 3..4 an algorithm for (approximate) determination of the optimal family without explicit use of the weight parameter λ. This means that a (suboptimal) boundary curve is determined in a finite number of steps, and without the need to carry an exhaustive search in λ.

3..1 Efficient Scale-Recursive Implementation

Since the subband-domain optimal filter has multichannel inputs and outputs, scale-recursiveness is, in fact, synonymous with channel-recursiveness. Channel-recursive optimal, as well as adaptive, schemes (for single-rate channels) have been constructed, for instance, by Ling and Proakis [7], and by Lev-Ari [8]. However, such schemes are not directly applicable to *nonuniform multirate processing*, i.e., to the case when the decimation ratios D_i of Fig. 7 are not all the same. This happens, for instance, in the context of wavelet-based estimation. This difficulty can be approached in a variety of ways including, in particular, techniques that result in uniform decimation:

- Using a so-called *wave-packet* configuration [9, 10], which results in equal decimation ratios D_i in all channels (Fig. 9). In contrast, the better known wavelet configuration results in a binary dilation relation between the decimation ratios, i.e., $D_i = 2D_{i-1}$ for all i.
- Splitting higher-rate channels into multiple lower-rate channels via time-multiplexing, as shown in Fig. 14 for the wavelet configuration.

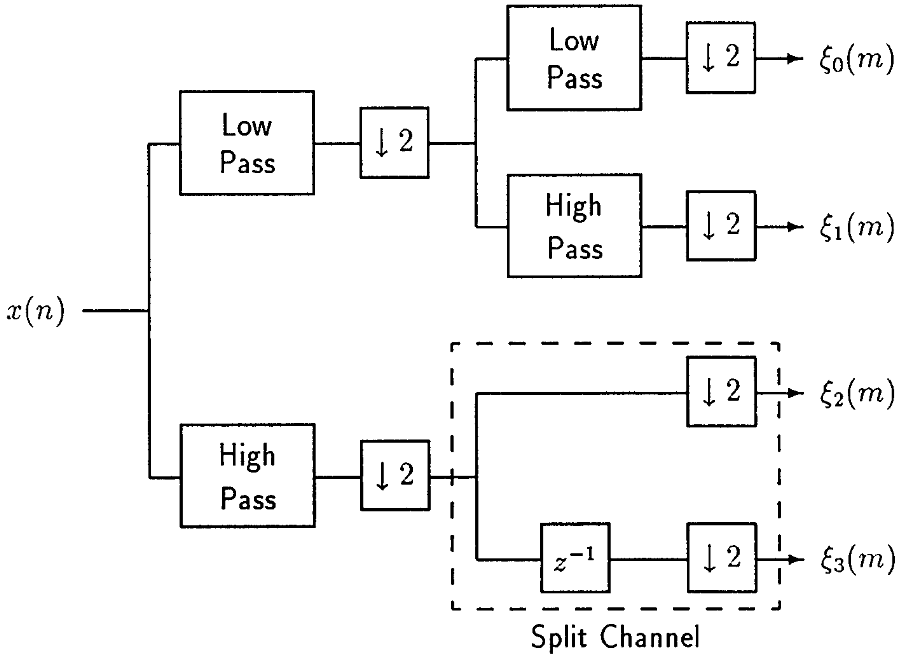

Figure 14: Wavelet configuration of filter bank, with channel splitting

The advantages and disadvantages of each approach will be the subject of further research. In this report we consider only the issue of statistical coupling between channels: using the wave-packet configuration makes it possible to minimize this coupling and, consequently, to achieve significant reduction in implementation cost.

In general, the introduction of multichannel subband-domain processing schemes results in a mild increase of the computational cost. For instance, while the implementation of an M-th order input-domain FIR filter requires $\mathcal{O}(M)$ computations per data sample, the implementation of the equivalent subband-domain filter requires $\mathcal{O}(L^2 \cdot \frac{M+2M_H}{L}) = \mathcal{O}\big(L \cdot (M+2M_H)\big)$ computations (where L is the number of bands). This is so because the computational cost of multichannel estimation is proportional both to the filter order, and to the *square of the number of channels.* However, since the subband rate is slower (by a factor of L) than the rate of the input data, we conclude that the computational cost of optimal (or adaptive) FIR filtering in the subband domain is $\mathcal{O}(M + 2M_H)$ per **input data sample**. The same orders of magnitude apply also to the implementation of *adaptive* multichannel FIR filters [8]. Thus, subband domain processing is comparable to input domain processing, both in terms of the achievable performance (which is exactly the same for both techniques), and the attendant computational cost (which is approximately the same).

3..1.1 Inter-Channel Correlation and Overlap

Scale-recursive optimal filtering is particularly attractive because of its enhanced ability to trade-off performance for computational cost, without altering the filter order. As explained earlier, such a trade-off can be achieved by using a partial subset of the subband domain channels, resulting in a dramatic reduction in computational cost. The attendant degradation in performance can be kept relatively small by designing the filter bank so as to minimize statistical coupling between channels.

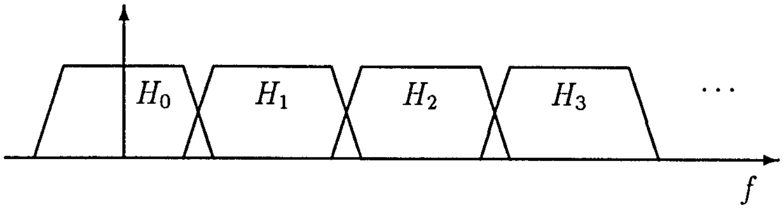

Figure 15: Filter bank with negligible overlap between non-adjacent subband channels.

Indeed, by choosing the filters $H_i(z)$ in such a way that $H_i(e^{j2\pi f})$ and $H_j(e^{j2\pi f})$ have negligible overlap for all $|i-j| \geq 2$ (Fig. 15), we find that

$$E\{\xi_i(k)\,\xi_j^*(l)\} \approx 0 \quad , \quad E\{\delta_i(k)\,\xi_j^*(l)\} \approx 0 \qquad \text{for } |i-j| \geq 2 \tag{42}$$

for all k, l. This qualitative statement relies on the observation that the correlation between subband channels can be expressed in terms of frequency domain integrals that involve the product $H_i(e^{j2\pi f})H_j^*(e^{j2\pi f})$, which is assumed to be negligible for $|i-j| \geq 2$.

A quantitative statement about the relation between channel overlap and correlation of subband signals must involve the notion of *channel overlap coefficients*, which we have defined as

$$\rho_{ij} \;:=\; \sqrt{\frac{\int_0^1 \, |H_i(e^{j2\pi f})|^2 \, |H_j(e^{j2\pi f})|^2 \; df}{\|H_i\|^2 \, \|H_j\|^2}} \tag{43}$$

where $\|H_i\|^2 \;:=\; \int_0^1 \, |H_i(e^{j2\pi f})|^2 \; df$. Thus, we say that the overlap between non-adjacent channels is negligible if the corresponding overlap coefficients are all small, viz.,

$$\rho_{ij} \ll 1 \qquad \text{for all} \quad |i-j| \geq 2 \tag{44}$$

We have shown [1] that

$$\Big|\, E\,\{\,\xi_i(k)\,\xi_j^*(l)\,\}\,\Big| \;\leq\; \rho_{ij}\; \|H_i\|\; \|H_j\|\; E\,|x(n)|^2$$

and

$$\Big|\, E\,\{\,\delta_i(k)\,\xi_j^*(l)\,\}\,\Big| \;\leq\; \rho_{ij}\; \|H_i\|\; \|H_j\|\; \sqrt{E\,|d(n)|^2 \,\cdot\, E\,|x(n)|^2}$$

In particular, if the filter-bank corresponds to a wave-packet configuration, as in Fig. 9, then $\|H_i\| := \|H\| = 1$ for all i, so that

$$\Big|\, E\,\{\,\xi_i(k)\,\xi_j^*(l)\,\}\,\Big| \;\leq\; \rho_{ij}\; E\,|x(n)|^2$$

Thus, reducing the inter-channel overlap indeed results in suppression of the correlation between subband channels.

3..1.2 Diagonal and Tridiagonal Configurations

Focusing on $\widehat{\delta}_i$, the estimate of a single subband component of $d(\cdot)$, we observe that under the assumption (44), namely $\rho_{ij} \ll 1$ for all $|i-j| \geq 2$, the linear least squares estimate of $\delta_i(m)$ depends mostly on the three signals $\xi_{i-1}(\cdot)$, $\xi_i(\cdot)$, and $\xi_{i+1}(\cdot)$, while the contribution from the remaining elements of $\boldsymbol{\xi}(\cdot)$ is negligible. Constructing an estimate of $\delta_i(m)$ that is based solely on these three components of $\boldsymbol{\xi}(\cdot)$ results in an order of magnitude reduction in computational cost. The small degradation in the quality of the estimate depends on the values of ρ_{ij} for $|i-j|>2$ and can be minimized by appropriate design of the filter bank. The resulting subband-domain filter $\mathcal{T}$ is called *tridiagonal*, because the estimate $\widehat{\delta}_i$ depends only on the three subband channels ξ_{i-1}, ξ_i, and ξ_{i+1} (Fig. 16).

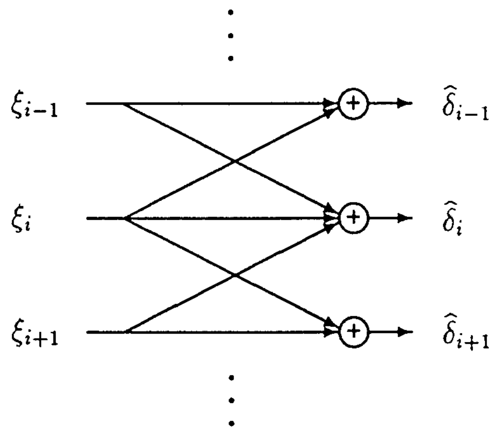

Figure 16: The structure of a tridiagonal subband-domain filter.

A further simplification of the filtering configuration involves using a *diagonal* subband-domain filter $\mathcal{T}$, as shown in Fig. 17.

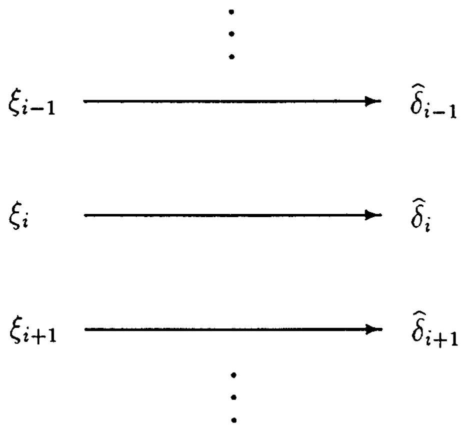

Figure 17: The structure of a diagonal subband-domain filter.

The corresponding computational cost of implementing a diagonal or tridiagonal subband-domain filter now becomes $\mathcal{O}(L \cdot \frac{M+2M_H}{L}) = \mathcal{O}(M + 2M_H)$ computations per subband-domain sample, or $\mathcal{O}(\frac{M+2M_H}{L})$ *per input sample.* In addition to this order-of-magnitude savings in computational cost, using subband-domain filtering makes it possible to achieve further cost reduction by selecting a subset of the available L subband-domain channels. This may come at the expense of some additional degradation in performance.

3..1.3 Structure of the Subband-Domain Estimator

The construction of a (suboptimal) subband-domain estimation scheme involves the selection of five design parameters, corresponding to the five factors that affect the cost-performance trade-off discussed in the introduction. These five parameters are:

- The number of subband channels L, which determines the resolution level.
- The transfer function of the filter-bank, which determines the degree of channel overlap.
- The utilization ratio η.
- The inter-channel estimation order M_1, which determines the estimation sparsity.
- The intra-channel estimation order M_2, which is the number of samples from each subband channel used in estimation.

The inter-channel and intra-channel orders can be, in general, different from channel to channel. However, here we consider only the uniform case, where all channels use the same inter- and intra-channel orders.

In particular, we focus on filter-banks associated with the notion of wavelets and wave-packets. Such filter-banks are constructed using a binary tree configuration which involves a single undetermined prototype lowpass filter $P(z)$ [9]. Thus, the resolution level of such filter-banks is determined only by the depth of the binary tree, while the degree of channel overlap depends only on the response of the prototype filter $P(z)$.

We use an r-layer wave-packet-based filter-bank (Fig. 10), so that the total number of available subband channels is $L = 2^r$. We demonstrate the effect of utilization ratio by processing only the subset of all available channels corresponding to $1 \leq i \leq \eta L$, where η can vary in the range $1/L \leq \eta \leq 1$. We consider both inter-channel and intra-channel sparsity. The first type is represented by two sparse estimation schemes:

- Diagonal estimation, in which the estimate of $\delta_i(\cdot)$ is constructed from samples of $\xi_i(\cdot)$ only $(M_1 = 1)$.
- Tridiagonal estimation, in which the estimate of $\delta_i(\cdot)$ is constructed from samples of $\xi_i(\cdot)$, $\xi_{i-1}(\cdot)$ and $\xi_{i+1}(\cdot)$ $(M_1 = 3)$.

Together with non-sparse estimation (which uses all available channels) this gives rise to three choices of inter-channel order, which we call "full", "tridiagonal", and "diagonal." As for the intra-channel order, we consider only two cases: (i) full, which uses all $N/2^r$ samples available from a single channel, and (ii) memoryless, which uses a single sample ($M_2 = 1$).

We use a block-processing scheme, so that subband-domain processing is carried out at the block-rate which is $1/N$ of the input signal rate, where N is the length of the input data block (see Fig. 10). The total number of samples per subband channel (per one block of input data) is $N/2^r$ where r is the number of layers of our wavelet-based filter bank. The use of a block-processing scheme introduces another reduction in the overall implementation cost, since the number of computations per one sample of the input data is reduced by a factor of N.

3..2 Cost-Performance Trade-off

In the example we consider here, the signal $x(n)$ is generated by adding a white noise $w(n)$ to desired signal $d(n)$, while $d(n)$ is the output of a lowpass filter excited by another white noise which is uncorrelated to $w(n)$. We now turn to evaluate the estimation error for our example as a function of the various design parameters. We scale the estimation error by the (square root of) the power of the desired signal $d(n)$, so that the resulting relative error can be compared across examples.

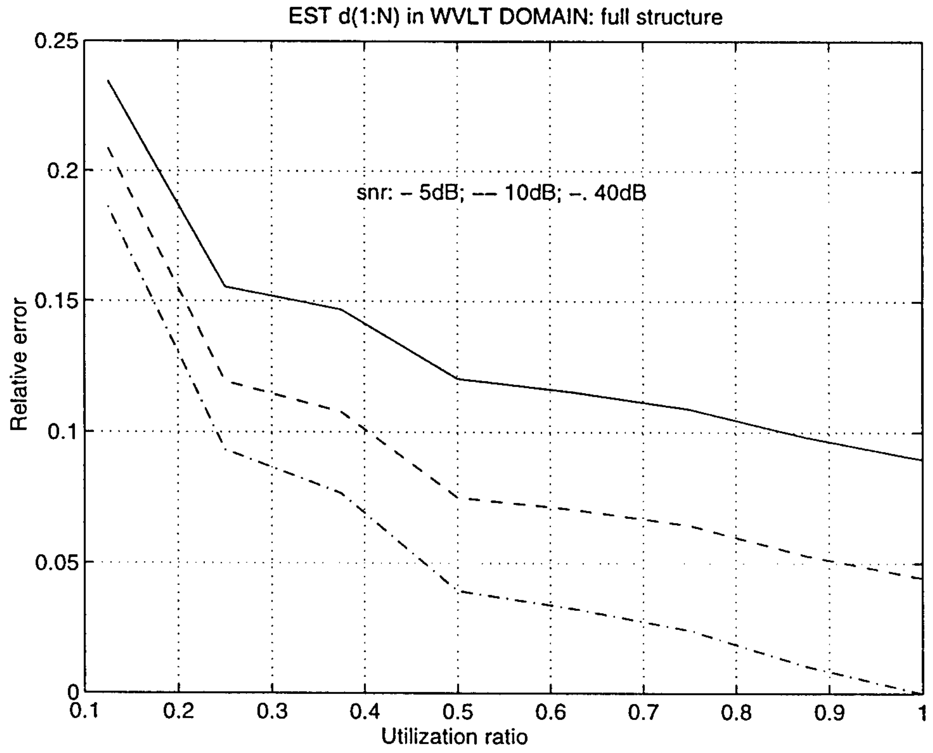

Figure 18: Effect of utilization ratio

3..2.1 Effects of Utilization Ratio (η)

The effect of the utilization ratio η alone is shown in Fig. 18, which involves three different levels of the signal-to-noise ratio (SNR) in the received signal $x(n)$. The error reduces with increasing η, the effect being more pronounced for high SNRs. This means that we can trade-off performance (= estimation error) for cost ($\sim$ utilization ratio). An explicit demonstration of this trade-off is provided in Fig. 11, where each curve corresponds to varying the value of η.

3..2.2 Effects of Estimation Sparsity

The introduction of inter-channel sparsity into our estimation schemes results in very minor degradation in performance, except when η is close to unity (Fig. 19). However, since sparse estimation requires fewer computations, the cost-performance trade-off curves (Fig. 11) show a dramatic advantage for sparse schemes and, in particular, for the diagonal configuration: one can achieve a reduction of up to an order of magnitude (i.e., a factor of 10) in computational cost for a given level of estimation error.

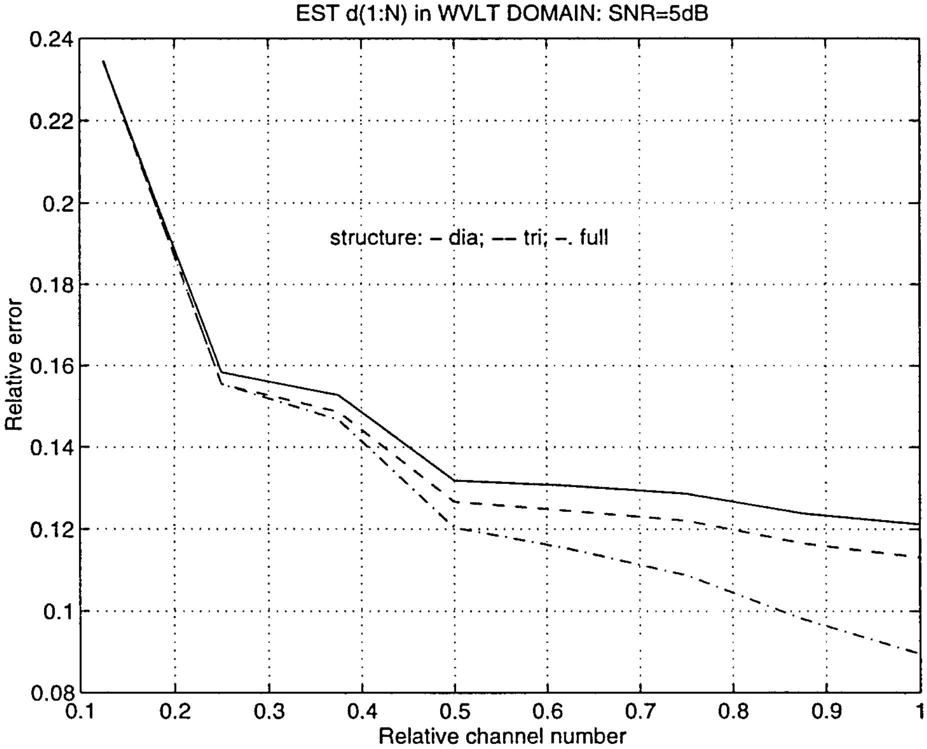

Figure 19: Effect of inter-channel sparsity

The effect of intra-channel sparsity (i.e., filter order) is similar to that of inter-channel sparsity (Fig. 20). Again, using less samples results in an improved cost-performance trade-off (Fig. 12), with up to 2 orders of magnitude reduction in computational cost.

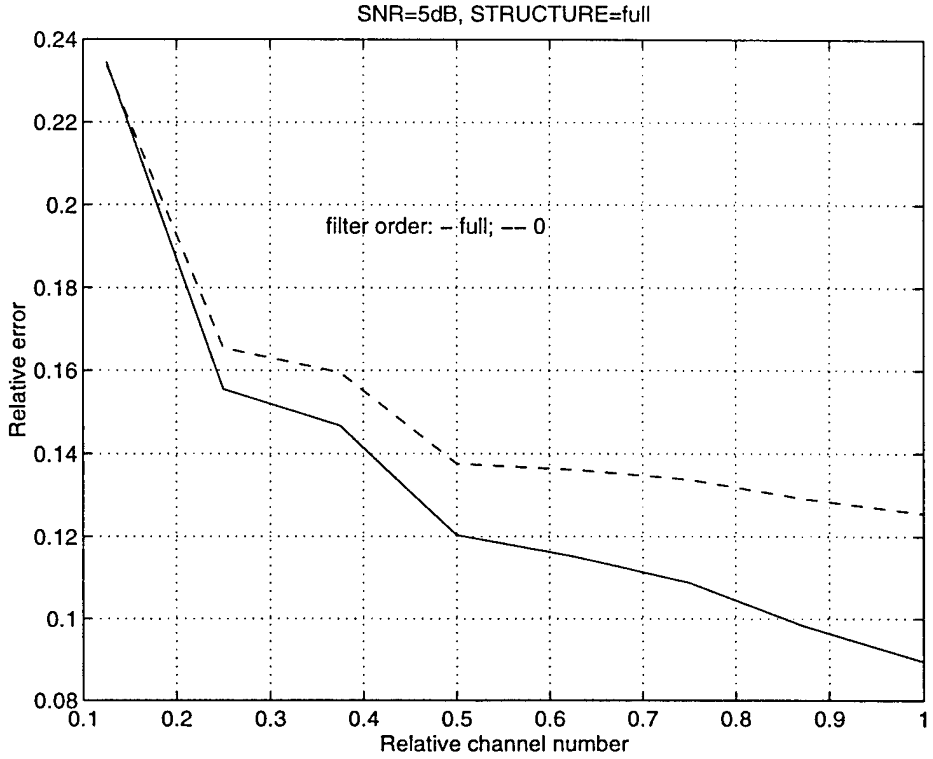

Figure 20: Effect of intra-channel sparsity

3..2.3 Effects of Resolution Level

Our block processing scheme, in which the overall number of samples remains fixed across all resolution levels, results in estimation errors that are essentially invariant with respect to the level of resolution (Fig. 21). However, the introduction of cost considerations complicates our analysis: for a full (i.e., non-sparse) configuration, the cost of estimation reduces with r (the number of layers), while the cost of subband decomposition increases with r. On the other hand, for a diagonal (i.e., maximally-sparse) configuration, the cost of estimation remains invariant with r, but the performance should slightly degrade with increasing r. Both cases suggest the existence of an optimal choice for r in each particular instance: further analysis is needed to determine this optimal choice.

3..2.4 Effects of Wavelet Choice

One example of the effects of filter-bank selection is shown in Fig. 22, where the use of two different Daubechies wavelets is compared. While our example shows a definite preference for simpler filter banks, more research is required in order to determine the best choice of a filter bank for a given estimation problem.

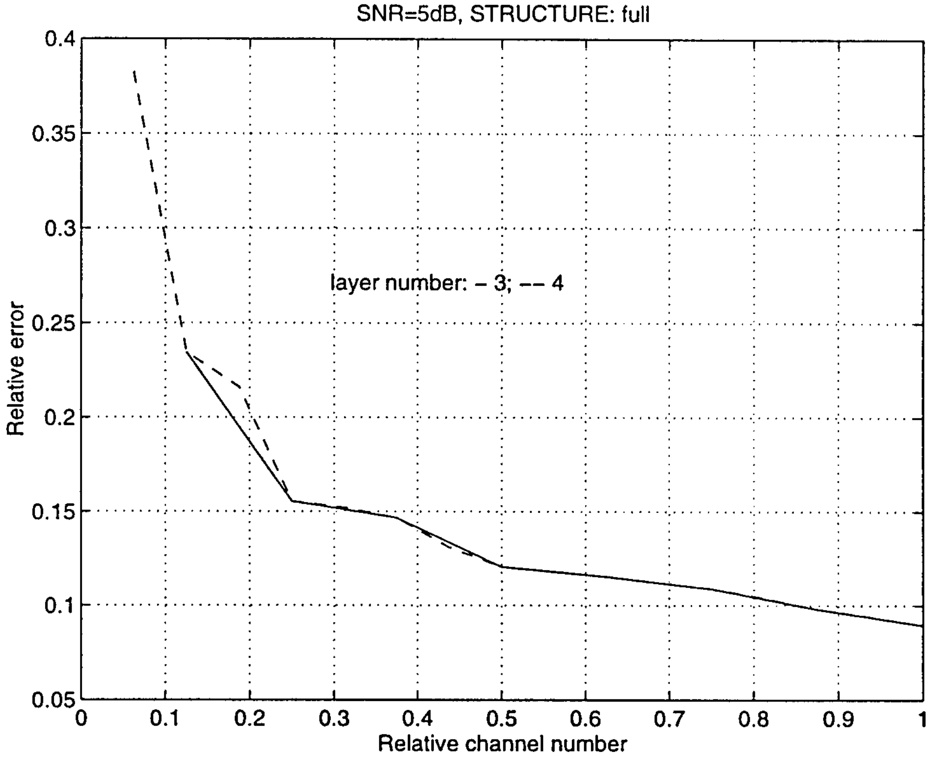

Figure 21: Effect of resolution level

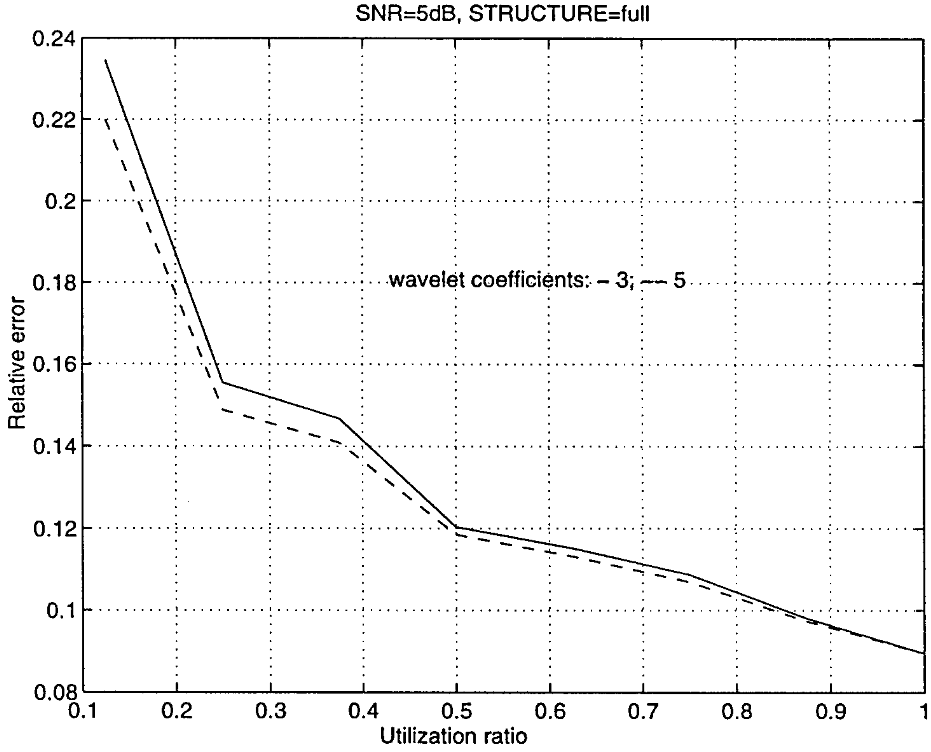

Figure 22: Effect of wavelet choice: comparing the use of Daubechies wavelets of order 3 and 5.

3..3 Optimal Filter Bank Configuration

Our analysis of the estimation error associated with a given sparse estimation scheme [1] has resulted in closed-form expressions which we could apply to the problem of optimal subtree selection [11]. In particular, we have shown that the estimation error associated with the memoryless, diagonal sparse estimation scheme described in [1] can be expressed in the form

$$\mathcal{E} = \mathcal{E}_0 + \sum_i \Delta\mathcal{E}_i \quad , \qquad \Delta\mathcal{E}_i > 0$$

where the summation is over all the vertices ("binary splits") of the subtree, and where $\Delta\mathcal{E}_i$ depends only upon the power spectrum of the signal at the node i. Similarly, we have shown that the computational cost associated with each subtree configuration can also be expressed in the form

$$\mathcal{C} = \mathcal{C}_0 - \sum_i \Delta\mathcal{C}_i \quad , \qquad \Delta\mathcal{C}_i > 0$$

where again, the summation is over all the vertices of the subtree.

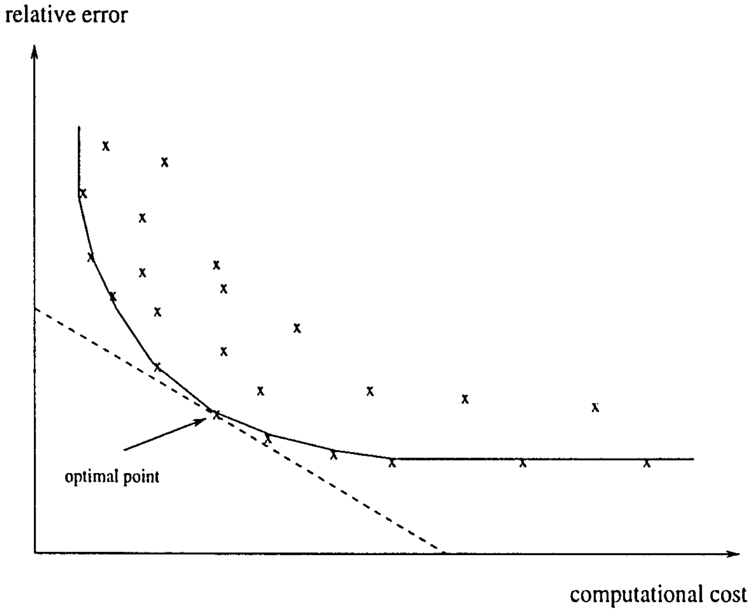

Figure 23: The cost-performance chart and the optimal boundary curve

The pair $(\mathcal{C}_0, \mathcal{E}_0)$ characterizes the empty subtree, corresponding to processing in the input domain, i.e., no filter-bank is being used. This is the rightmost point on the optimal (boundary) curve, corresponding to $\lambda = 0$. As the subtree grows to include more and more vertices, the computational cost reduces and the estimation error increases, resulting in $(\mathcal{C}, \mathcal{E})$ points to the left, and above, the point $(\mathcal{C}_0, \mathcal{E}_0)$ (Fig. 23). The foregoing discussion implies that we are faced with a graph optimization problem, where with each vertex we associate an (ordered) pair of positive incremental costs, namely $(\Delta\mathcal{C}_i, \Delta\mathcal{E}_i)$, such that the overall cost associated with a given subtree is

$$J(\lambda) \;=\; \lambda\mathcal{C} + (1-\lambda)\,\mathcal{E} \;=\; \left[\lambda\mathcal{C}_0 + (1-\lambda)\,\mathcal{E}_0\right] \;+\; \sum_{i \epsilon I} \left[-\lambda\Delta\mathcal{C}_i + (1-\lambda)\,\Delta\mathcal{E}_i\right] \qquad (45)$$

The optimization (for a given λ) is carried out with respect to I, the set of vertices that determines the subtree.

Every specific subtree of the tree-structured filter-bank of Fig. 9 results in a point in the $\mathcal{C}$–$\mathcal{E}$ chart (Fig. 23). The collection of all possible subtrees determines a region in this chart whose piecewise-linear lower boundary curve is up-convex. The optimal subtree for a given λ is that boundary point where the slope of the boundary curve equals $-\lambda/(1-\lambda)$ (Fig. 23). This is so because the tangent to the curve at this point is characterized by

$$dJ \;=\; \lambda\, d\mathcal{C} + (1-\lambda)\, d\mathcal{E} \;=\; 0 \qquad \longrightarrow \qquad \frac{d\mathcal{E}}{d\mathcal{C}} = -\frac{\lambda}{1-\lambda}$$

We have constructed an algorithm that determines the optimal subtree for a given (fixed) value of the weight coefficient λ [11], as described below.

Our algorithm is based on the observation that the cost function $J(\lambda)$ of (45) is additive with respect to the vertices of the subtree. Also we restrict our search to subtrees of the complete binary tree of a fixed depth (say r), so as to limit the set of vertices over which we need to carry out our search. Thus, the optimal subtree can be determined by *pruning*, i.e., by starting with a complete binary tree (of depth r), and removing subsets of vertices for which $\Delta J < 0$. To be more specific, the only vertex subsets considered for removal are of the form

$$V_k = \text{set of vertices that are the children of vertex } k \tag{46}$$

Thus, at any given stage of our pruning procedure we look for a vertex k for which

$$\sum_{i \epsilon V_k} \Delta J_i < 0$$

The search begins with the leaves of the subtree and continues, layer by layer, to vertices further away from the leaves, until one vertex with $\sum \Delta J_i < 0$ is found. Having found a vertex like that we remove (prune) all vertices in V_k and repeat our search procedure. The procedure terminates when $\sum \Delta J_i \geq 0$ for every subset of vertices V_k as defined in (46). The resulting subtree is a member of the optimal family and determines a point on the boundary curve in the $\mathcal{C}$–$\mathcal{E}$ chart.

3..4 A "Greedy Type" Algorithm

In order to avoid the need to search over the entire range of λ we introduce here an algorithm that attempts to directly determine the corner points of the (piecewise-linear) optimal boundary curve. We know that the endpoints of this curve are: (i) the empty subtree, which is optimal for $\lambda = 0$, and yields the lowest error $\mathcal{E}$; and (ii) the complete binary tree of infinite depth (i.e., $r = \infty$), which is optimal for $\lambda = 1$, and yields the lowest cost $\mathcal{C}$. Thus, our algorithm starts with the empty subtree and attempts to determine the remaining corner points one-by-one, in the order of their appearance on the boundary curve.

To achieve this goal we increment our subtree at each step by a single vertex in such a way that the ratio $\frac{\Delta\mathcal{E}_i}{\Delta\mathcal{C}_i}$ associated with incorporating vertex i into the subtree is as small as possible.

Thus, our algorithm is locally-optimal, i.e., it is a "greedy type" algorithm. It produces a sequence of subtrees where each member in the sequence is a subset of its immediate successor, and differs from it by a single vertex. The piecewise-linear curve obtained by connecting the $\mathcal{C}$–$\mathcal{E}$ points that represent this sequence of subtrees is not necessarily convex (or optimal). Thus, an improved curve can be obtained by selecting the convex hull of the points determined by our greedy type algorithm (Fig. 24). This still yields a suboptimal sequence of subtrees.

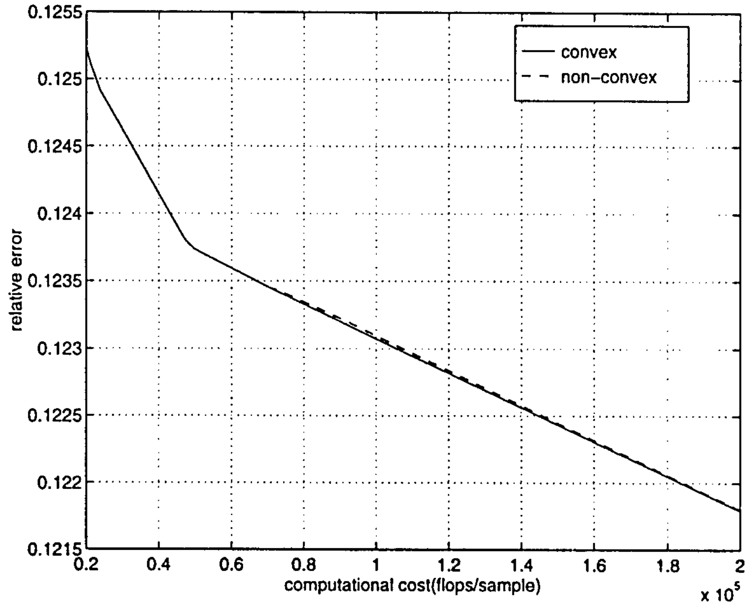

Figure 24: Cost-performance tradeoff: the suboptimal curve. Solid line: the convex hull; dashed line - the output of the greedy-type algorithm.

References

[1] H. Liu and H. Lev-Ari, "Optimal Filtering in the Subband Domain," CDSP Technical Report No. TR-CDSP-94-32, Northeastern University, Dec. 1994. See also *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 2076-2079, Detroit, MI, May 1995.

[2] A. Gilloire and M. Vetterli, "Adaptive Filtering in Subbands with Critical Sampling: Analysis, Experiments, and Application to Acoustic Echo Cancellation," *IEEE Trans. Sig. Proc.*, Vol. 40, pp. 1862-1875, Aug. 1992.

[3] R.R. Coifman and M.V. Wickerhauser "Entropy-Based Algorithms for Best Basis Selection," *IEEE Trans. Information Theory*, Vol. 38, pp. 713-718, Mar. 1992

[4] A.H. Tewfik, D. Sinha, and P. Jorgensen, "On the Optimal Choice of a Wavelet for Signal Representation," *IEEE Trans. Information Theory*, Vol. 38, pp. 747-765, Mar. 1992.

[5] S.G. Mallat and Z. Zhang, "Matching Pursuits with Time-Frequency Dictionaries," *IEEE Trans. Sig. Proc.*, Vol. 41, pp. 3397-3415, Dec. 1993.

[6] C.W. Therrien, *Discrete Random Signals and Statistical Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1992.

[7] F. Ling and J.G. Proakis, "A Generalized Multichannel Least Squares Lattice Algorithm Based on Sequential Processing Stages," *IEEE Trans. Acoust. Speech and Signal Processing*, Vol. ASSP-32, pp. 381-390, Apr. 1984.

[8] H. Lev-Ari, "Modular Architectures for Adaptive Multichannel Lattice Algorithms," *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol. ASSP-35, pp. 543-552, Apr. 1987.

[9] P.P. Vaidyanathan, *Multirate Systems and Filter Banks*, Prentice-Hall, Englewood Cliffs, NJ, 1993.

[10] M.V. Wickerhauser, "Lectures on Wavelet Packet Algorithms," *Proceedings of INRIA Workshop on "Ondelettes et Paquets d'Ondelettes"*, pp. 31-99, Roquencourt, France, June 1991.

[11] H. Liu and H. Lev-Ari, "Optimized Filter Banks for Efficient Subband-Domain Estimation," CDSP Technical Report No. TR-CDSP-96-38, Northeastern University, Feb. 1996.

Graduate Students

Hong Liu

Publications

H. Liu and H. Lev-Ari, "Scale-Recursive Optimal Filtering," CDSP Technical Report No. TR-CDSP-94-25, Northeastern University, Dec. 1994.

H. Liu and H. Lev-Ari, "Optimal Filtering in the Subband-Domain," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 2076-2079, Detroit, MI, May 1995.

H. Liu and H. Lev-Ari, "Optimized Filter Banks for Efficient Subband-Domain Estimation," CDSP Technical Report No. TR-CDSP-96-38, Northeastern University, Feb. 1996.

4. Parallel Algorithms and Architectures for Discrete Wavelet Transforms – Elias S. Manolakos

In this section we provide a summary of the main results obtained by Prof. Elias S. Manolakos and his graduate students on the *synthesis* of parallel computational structures for the Discrete Wavelet Transform (DWT). Improving algorithm performance through parallelism has been addressed from two points of view: By optimizing aggregate throughput in a series of problem instances by means of modular VLSI *architectures* having a small number of processing elements (PEs); and by means of *scalable algorithms* for parallel supercomputers having a large number of PEs and a fixed interconnection network.

Our early results on the synthesis of VLSI architectures for the DWT appeared and in the Proceedings of the *SPIE Conf. on Mathematical Imaging: Wavelet Applications in Signal and Image Processing* [1] and in *VLSI Signal Processing VI* [2]. More detailed journal papers also including more recent work both on the synthesis methodology introduced as well as the resulting DWT array solutions are currently under review in the *IEEE Transactions on Signal Processing* [5, 6]

Our results on the parallel scalable DWT algorithms for mesh and hypercube networks have been accepted for publication to the special issue on wavelet based signal processing of the *Journal on Multidimensional Signal Processing* [7] and will appear in 1996. Preprints of all recently accepted publications are available on the Web coordinates: http://www.cdsp.neu.edu/info/manolakos.html.

To remain focused, in the sequel we only highlight the main points of our research findings during the whole three year period of the grant. The technical details can be found in the above mentioned refereed publications that are available, and have also been discussed in the submitted annual reports (April 1994, April 1995).

4..1 Synthesis of VLSI Architectures for Discrete Wavelet Transforms

Our main objective was twofold: to understand and analyze the parallelism inherent in the DWT algorithm, and to synthesize parallel processing structures for its computation that could be implemented in VLSI. The architectures derived have:

- Regular structure and $O(L)$ Processing Elements (PEs), where L is the number of wavelet filter coefficients (usually in the range of 4-12).
- Minimal amount of memory distributed to the processors and no global memory or interprocessor interconnection networks that limit architectural scalability.
- Simple control distributed to the processors.

A new methodology for synthesizing distributed memory and control parallel computational structures has been developed. It is suitable for signal processing and numerical algorithms with partially irregular data dependencies structure, such as the DWT, that cannot be described by a set of *Uniform* or *Affine Recurrence Equations* (UREs, AREs). For this class of algorithms it is known that conventional linear space-time mapping methods fail to provide efficient parallel array solutions. Our proposed methodology is based on constructing and applying the appropriate *non-linear* index space transformations (re-indexing) that lead to a reformulation of the given algorithm in two dimensions (i.e. using only two indices) so that after applying conventional linear space-time mapping very efficient linear processor arrays suitable for VLSI implementation are derived. So it can be though of as the appropriate "pre-processing" required to bring the algorithm in a more benign form at which point the arsenal of linear mapping theory with all its known advantages can be called to complete the synthesis successfully.

Several families of systolic and semi-systolic arrays for computing general (any number of octaves) DWTs have been synthesized in that way. To the best of our knowledge they are among the most efficient $O(L)$-processor solutions available in the literature, as it is also acknowledged by others (see for example [3] page 307). By using our formal and top-down algorithm-to-architectures synthesis approach we have been able to:

- derive behavioral models of processor arrays that can be easily translated to VHDL code
- prove analytically the correctness of the designs at the behavioral level before building any hardware
- compare/contrast deferent candidate designs in terms of latency (parallel running time), processors utilization and throughput

In the interest of space we will not re-iterate here on results already included in previous annual reports or more recent publications. For example, in the April 1994 report we presented three scalable array architectures which compute the DWT algorithm of an M-sample sequence and up to any desired octave J. They have been named SYST, BCAST and PIPE respectively and have been show to achieve latency of $3M/2 + 2(2)^{J-2} - 2$, $M + 2^{J-1} + 3$, and $M/2 + 2^{J-2} - 2$ respectively (where $M \geq 2^J$ is the input sequence length, L is the number wavelet filter coefficients). Therefore BCAST is 33% faster than SYST, and PIPE is 66% faster than SYST. Processor utilization for SYST, BCAST and PIPE is 66%, 100% and 100% respectively.

The systematic dependence analysis and localization of the algorithm and a 3rd-octave array solutions synthesis has appeared in the Proceedings of the *SPIE Conf. on Mathematical Imaging: Wavelet Applications in Signal and Image Processing* [1] . The more general case of arrays for any desired number of octaves J and sequences of length $M \geq 2^J$ have been published in *VLSI Signal Processing VI* [2]. More detailed journal papers including both the methodology introduced and the resulting array solutions have been submitted and are currently under review in the prestigious *IEEE Transactions on Signal Processing* [5, 6]

4..2 Scalable Parallel Algorithms for Wavelet Transforrms on the mesh and hypercube networks

We have also derived several parallel algorithms for 1-D and 2-D Discrete Wavelet Transforms that exhibit very good scalability properties on different general purpose processor networks such as the

2-D mesh and the *hypercube*. For example our data parallel SIMD algorithms can perform a J levels multiresolution analysis of a length M 1-D data sequence in $O(JL)$ time on an array of M PE's, where L is the number of wavelet filter coefficients. For instance, a $J = 10$ levels pyramid DWT decomposition of a length $M = 4096$ sequence can be completed in 20.7msec on the MasPar-1 2-D torus array [4].

We have primarily focused on parallel algorithms for the 2-D DWT for performing efficiently multiresolution analysis of large size SAR images. We considered two slightly different versions of the 2-D DWT, known in the literature as the *Standard* (S-, or Beylkin's) and *Non-standard* (NS-, or Mallat's) forms, that are both extensively used in different applications. We mapped both forms onto P processors under two data partitioning schemes of the input matrix $M \times M$, namely *checkerboard* (-CP) and *stripped* (-SP) partitioning. We are introducing a precise performance model for the four parallel algorithmic 2-D DWT variants in order to investigate their *scalability* properties on the widely used Mesh and Hypercube P-processor networks with *store-and-forward* as well as *cut-through* routing. By scalability here we mean the ability of a parallel algorithm for making efficient use of increasing computational resources (processors) on a given network.

We have derived the asymptotic *isoefficiency* function of the four algorithms, that provides a lower bound on the rate of growth of the input problem size (in $O(M^2)$) as a function of the processors P, necessary to maintain a constant level of efficiency. It has been found that the two checkerboard partitioned algorithms on the Hypercube and on the cut-through-routed Mesh are scalable as $M^2 = \Omega(P \log P)$ (Non-standard form, NS-CP), and as $M^2 = \Omega(P \log^2 P)$ (Standard form, S-CP); while on the store-and-forward-routed Mesh they are scalable as $M^2 = \Omega(P^{\frac{3}{3-\gamma}})$ (NS-CP), and as $M^2 = \Omega(P^{\frac{2}{2-\gamma}})$ (S-CP), where M^2 is the number of elements in the input matrix, and $\gamma \in (0, 1)$ is a parameter relating M to the number of desired octaves J in the wavelet decomposition, as $J = \lceil \gamma \log M \rceil$.

The Standard form algorithm with stripped partitioning (S-SP) is scalable on the Hypercube as $M^2 = \Omega(P^2)$, and it is *unscalable* on the CT-routed Mesh (does not assume an isoefficiency function). Although asymptotically the stripped partitioned algorithm S-SP on the Hypercube would appear to be inferior to its checkerboard counterpart S-CP, a detailed analysis based on the often ignored proportionality constants of the isoefficiency function indicates that S-SP may actually be *more* efficient that S-CP over a realistic range of machine and problem sizes. A milder form of this result holds on the Mesh, where S-SP would, asymptotically, appear to be altogether unscalable.

Since the proportionality constants in our performance model are expressed in terms of the computation speed $(1/t_c)$ as well as the communication parameters $(t_s, t_h, t_w$ the message startup, message header and per-word transfer times, respectively) and these parameters can be accurately estimated for a given algorithm/network pair, we can determine which algorithm/parallel machine combination is the most cost-effective candidate for performing 2-D DWT processing for ATR/D purposes, given the throughput requirements.

Our results have been accepted for publication to the special issue on Wavelet based signal processing of the *Journal on Multidimensional Signal Processing* [7] and will appear in 1996.

5. Model-Based Target Recognition by Parameter Estimation of Hierarchical Mixtures – Elias S. Manolakos

Target recognition was investigated in the framework of parameter estimation of hierarchical mixture densities by Prof. Elias S. Manolakos and graduate students. As a result, simple hierarchical classifiers have been developed for target/object recognition that have good detection capabilities even in the presence of substantial *target occlusion.*

A modified version of the Expectation-Maximization (EM) algorithm was formulated and used to derive recursive updating rules for the *a-posteriori* probabilities of a specific target being present in the image. The EM was extended to multiple level hierarchies in order to also deal with target translation and scale invariance. Regularizing constraints were introduced to allow reaching robust decisions in an unsupervised manner. A local and recurrent view for the clustering algorithm that can be easily parallelized was established after making certain approximations. The approach has the flexibility to account for the non-stationary statistics of the input data as well as non-additive noise models, two aspects that are very important for SAR based ATR.

Our simulation results suggest that the scheme works well even in the case of partially occluded targets presented at scales for which templates (models) are not available. The theoretical justification of this part of our work and the first simulation results with synthetic images have appeared in the Proc. of the *Conference on Information Systems and Sciences, Princeton*, March 1996 [8] and the April 1996 annual report. More recent results using natural images with rich background and occluded objects are accepted and will appear at the Proceedings of *IEEE International Conference on Image Processing*, October 1996 [9].

We present a brief summary of these latest results here as well. As an example of a real world situation and to test our scheme we considered the non-trivial recognition of traffic signs in complicated background. Since the background has no template, as a first approach, we tried to model it crudely as an average intensity. This can be very unreliable when the background has large variations. Sometimes we use two levels: a light and a dark level to model the background. In any case we are faced with a situation where our model for the objects in the image is highly reliable whereas the model for the background in very unreliable. We account for this discrepancy in the level of our confidence in different models by introducing a "temperature" term in the error database used as prior knowledge. The function of the temperature in our case is similar to its function in Monte Carlo simulations except that here we choose different temperature values for the objects/targets and the background(s) reflecting our different beliefs in their models. In particular we use a large temperature for the background relative to the objects.

Some indicative results obtained from simulations on real images are shown in Fig. 25. Also Fig. 26 presents the results with a semi-synthetic image with a toy pony and a tool that occlude each other. It illustrates the ability of the scheme to identify partially occluded objects in high noise. In all these images, the regions with high intensity (white) refer to regions of high *a posteriori* probability of the particular object being present.

Our experiments with real and semi-synthetic images suggest that the scheme can be used to recognize objects in the presence of occlusion and complicated background. Furthermore the scheme

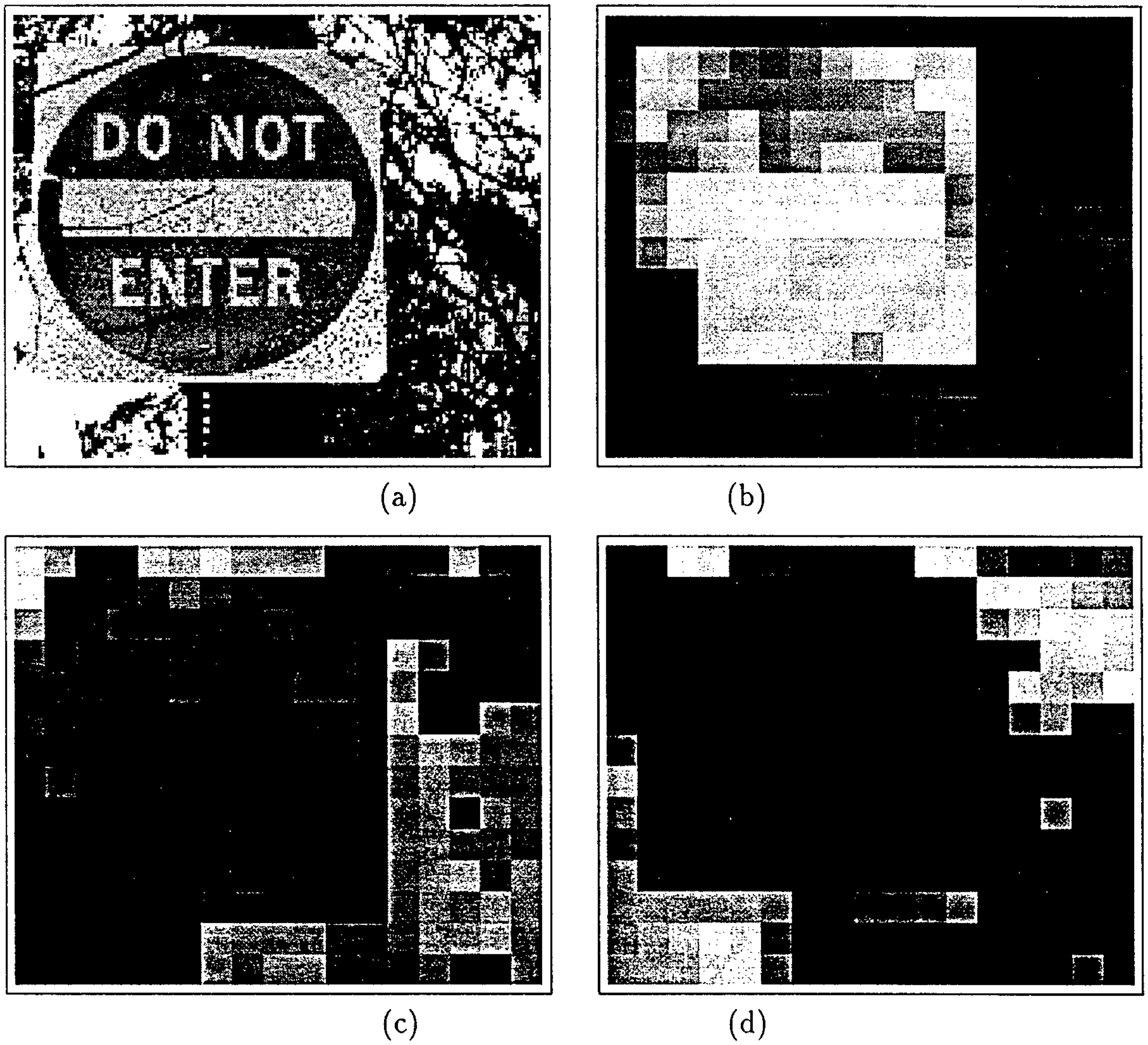

Figure 25: (a) Original Image, Posterior pdfs for (b) Traffic Sign, (c) Dark Background and (d) Light Background.

can handle different noise models. It is clear that a better model of the background will contribute towards improving the quality of recognition. The ideal values for free parameters such as the number of iterations of the EM loop, the size of the neighborhood and the value of the temperature of the background in relation to that of the objects must be determined through cross-validation.

We are currently working on the problem of reducing the complexity particularly in memory by implementing a hypothesis pruning technique based on the wavelet decomposition. We are also considering better modeling of the background with the available data. We are planning to use this approach for the the automatic recognition of ground targets from SAR images if we could get Xpatch generated target signatures. In this way we will be able to integrate wavelet based signal processing with simple hierarchical classifiers "attached" at the leaves of the wavelet tree for the purpose of flexible and efficient ATR.

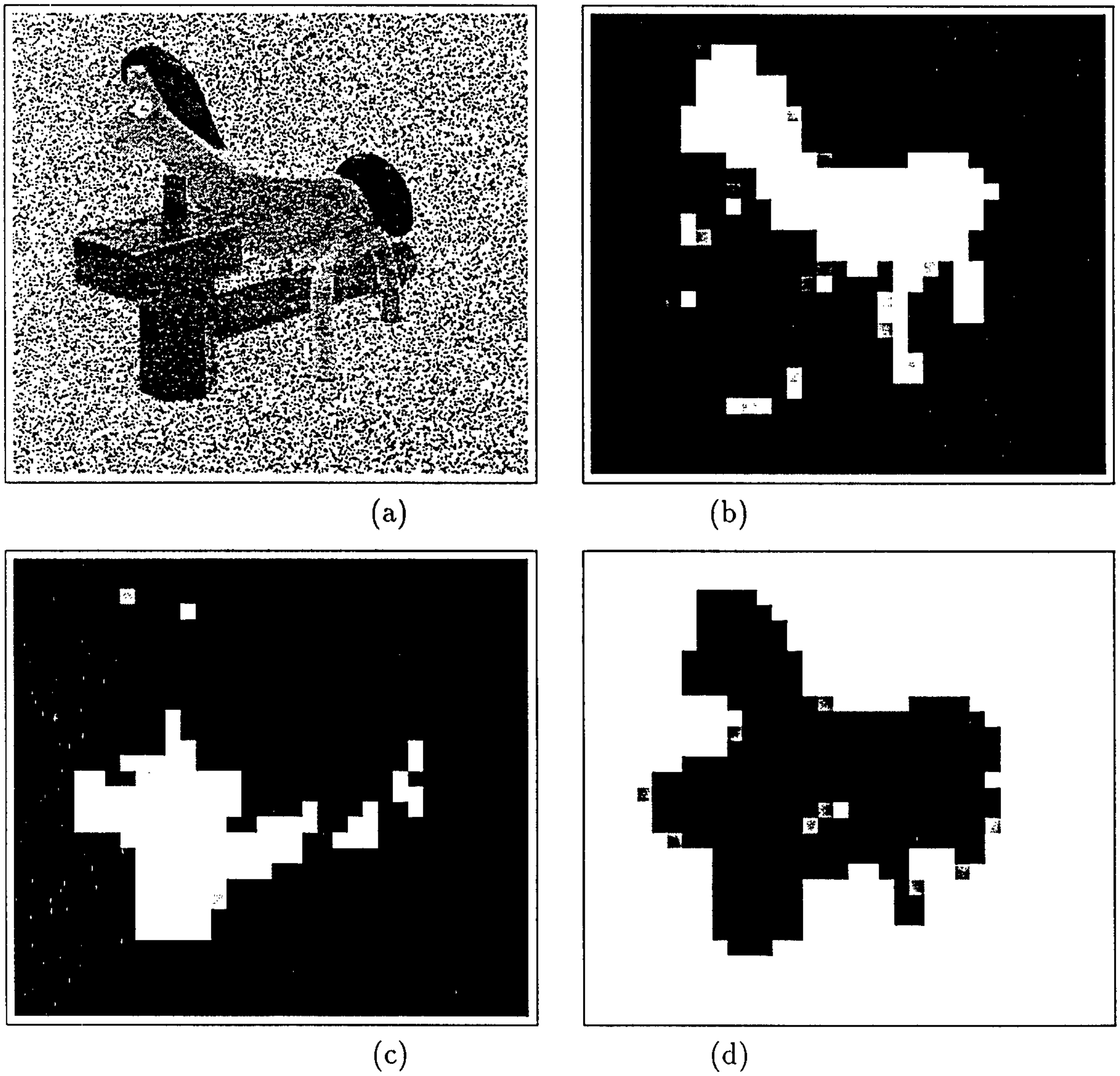

Figure 26: (a) Original noisy Image, Posterior pdfs for (b) Pokey, (c) Tool and (d) Background.

References

[1] J. Fridman and E. Manolakos, "On the Synthesis of regular VLSI architectures for the 1-D Discrete Wavelet Transform" in Proc. of *SPIE Conf. on Mathematical Imaging: Wavelet Applications in Signal and Image Processing II*, pp. 296-307, San Diego, July 1994.

[2] J. Fridman, E.S. Manolakos, "Distributed Memory and Control VLSI Architectures for the 1-D Discrete Wavelet Transform", in the Proc. *IEEE VLSI Signal Processing VII*, pp. 388-397, J. Rabaey, P. Chau, J Eldon editors, IEEE Press, Oct. 1994.

[3] J. Chen, M. Bayoumi, "A Scalable Systolic Array Architecture for the 2-D Discrete Wavelet Transforms", VLSI Signal Processing VIII, pp. 303-312, T. Nishitani, K. Parhi editors, IEEE Press, Oct. 1995.

[4] J. Fridman, E.S. Manolakos, "Discrete Wavelet Transform on a SIMD massively parallel platform", Proc. of the *International Conference on Signal Processing Applications and Technology*, (ICSPAT'95), Boston, submitted 1995.

[5] J. Fridman, E.S. Manolakos, "Parallel Processing for the 1-D Discrete Wavelet Transform. Part I: Algorithm Analysis and Localization" *IEEE Transactions on Signal Processing*, under review.

[6] J. Fridman, E.S. Manolakos, "Parallel Processing for the 1-D Discrete Wavelet Transform. Part II: Scalable Architectures Synthesis", *IEEE Transactions on Signal Processing*, under review.

[7] J. Fridman, E.S. Manolakos, "On the Scalability of 2-D Discrete Wavelet Transform Algorithms", *Multidimensional Systems and Signal Processing*, (special issue on Wavelets and Multiresolution Signal Processing), accepted to appear in 1996.

[8] V.P. Kumar and E.S. Manolakos. "Recurrent Architectures for Object Recognition by Parameter Estimation of Hierarchical Mixtures". In the Proc. of the *Conference on Information Systems and Sciences, Princeton*, March 1996.

[9] V.P. Kumar and E.S. Manolakos, "Unsupervised Model-Based Object Recognition by Parameter Estimation of Hierarchical Mixtures" to appear in the Proceedings of the *IEEE International Conference on Image Processing*, October 1996.

6. M.M. Kokar – Model-Theory Based Fusion Framework with Application to Wavelet-Features Based Mutisensor Target Recognition

In this report, we present a model-theory based fusion methodology for multisensor wavelet-features based recognition called Automatic Multisensor Feature-based Recognition System (AMFRS). The goal of this system is to increase accuracy of the commonly used wavelet-based recognition techniques by incorporating symbolic knowledge (symbolic features) about the domain and by utilizing a model-theory based fusion framework for multisensor feature selection.

Fusion is a process of combining (fusing) information from different sensors when there is no physical (fusing) law indicating the correct way to combine this information [7]. A fusion problem can be defined in terms of finding such a fusing law. Current solutions to the fusion problem consist of numerous clever, problem-specific algorithms [5, 11]. It is not clear what unifying theory exists behind all the fusion algorithms which would guide algorithm development. The lack of an unifying theory results in a difficulty in comparing various fusion solutions and often results in suboptimal solutions which negatively affect the accuracy of the multisensor recognition systems. Additionally, fusion often increases processing time. A computer system can be easily overloaded with data from different sensors to the degree that it cannot perform its task within the time constraints dictated by the application. Therefore, it has become important to develop an unifying approach for fusing/combining data from multiple sensors or from multiple measurements with minimal computational complexity.

One of the recent research efforts to develop a unifying fusion approach is described in [7], where a model-theory based Sensory Data Fusion System (SDFS) is proposed. The SDFS uses formal languages to describe the world and the sensing process. Models represent sensor data, operations on data, and relations among the data. Theories represent symbolic knowledge about the world and about the sensors. Fusion is treated as a goal-driven operation of combining languages, models, and theories, related to different sensors into one fused language, one fused model of the world and one fused theory. Kokar et al. [7] formalizes the main concept of the sensor data fusion theory in terms of models and theories. However, no algorithm for sensor data fusion is presented in [7], and the proposed sensor data fusion approach is not tested on real tasks and real noisy data.

In this work, we apply and extend the model-theory based SDFS to a multisensor wavelet-features based target recognition by using the AMFRS. The Discrete Wavelet Packet Decomposition (DWPD) is used for feature extraction, and the Best Discrimination Basis Algorithm (BDBA) and a domain theory for feature selection for each sensor. The BDBA [10] is based on a best basis selection technique proposed by Coifman and Wickerhauser in [3]. The BDBA selects the most discriminant basis. For the purpose of this work, the elements (wavelet coefficients) of the most discriminant basis are considered as features. Then, the domain theories for each sensor select interpretable wavelet features from the most discriminant basis. The wavelet features are interpretable in terms of symbolic target features. And finally, the wavelet-features are combined (fused) into one fused wavelet-features vector by utilizing the fused theory. The fused theory is derived from a symbolic knowledge about the domain and a relation between the symbolic knowledge and the most discriminant basis determined by using the BDBA. We show that the knowledge of

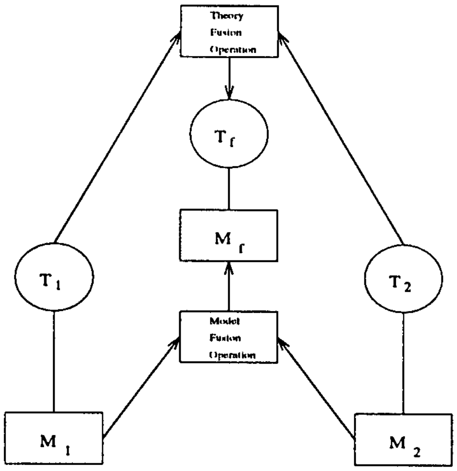

Figure 27: Model-Theory Based Fusion Framework

a fused theory allows us to select the combination of features (out of many possible combinations) which results in the increased recognition accuracy of the AMFRS. In particular, we show that the recognition accuracy of the proposed Automatic Multisensor Feature based Recognition System (AMFRS) is better than the recognition accuracy of a system that performs recognition using Most Discriminant Wavelet Coefficients (MDWC) as features. The AMFRS utilizes a model-theory framework (SDFS) for feature selection as described above, while MDWC are selected from the range and intensity most discriminant bases using a relative entropy measure [12].

6..1 Model-Theory Based Feature Fusion

In this work, we consider only two sensors for recognition. However, all results can be easily extended to more sensors without a loss of generality. Assume that n_1 features are selected from the measurement data of *Sensor 1* and n_2 features are selected from the measurement data of *Sensor 2*. The goal is to combine (fuse) these features into one feature vector consisting of exactly n features. Equivalently, we want to select n out of $n_1 + n_2$ features which gives us

$$C(n_1 + n_2, n) = \begin{pmatrix} n_1 + n_2 \\ n \end{pmatrix} = \frac{(n_1 + n_2)!}{(n_1 + n_2 - n)!n!} \tag{47}$$

possible combinations. This can be a big number. For example, for $n = n_1 = n_2 = 10$ we have 184,756 possible combinations. Therefore, we need a tool to select, from all these combinations, the combination of features which results in most accurate recognition decisions.

To select such a combination of features, we apply a model-theory based fusion framework shown in Fig. 27. In particular, the knowledge of a fused theory allows us to select the right combination of features. The fused theory is derived by using a theory fusion operator to fuse the theory T_1 for $Sensor_1$ and the theory T_2 for $Sensor_2$. There is no physical law known and therefore there is no theory fusion operator known for fusing data from these two sensors. Instead, we derive a theory fusion operator manually from a symbolic knowledge about the domain and a relation between the symbolic knowledge and features determined by using the BDBA. By applying the theory fusion

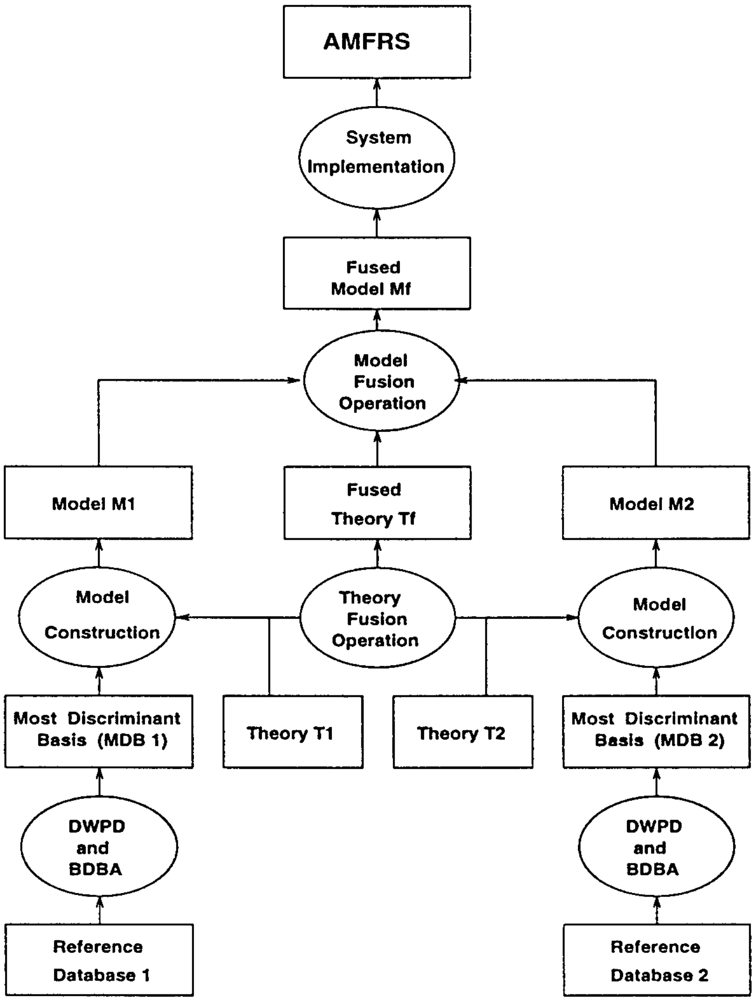

Figure 28: Feature Fusion for Automatic Multisensor Feature based Recognition System (AMFRS)

operator to the theory T_1 and the theory T_2, we derive the fused theory T_f. In the same manner, we derive manually a model fusion operator. By applying the model fusion operator to the model M_1 and the model M_2, we derive the fused model M_f.

Fig. 28 shows a block-diagram of steps involved in the design framework for Automatic Multisensor Feature-based Recognition System (AMFRS). The meaning of each block is the following:

1. *Reference Database 1* and *Reference Database 2* are databases of target/sound signatures.

2. *DWPD & BDBA* selects the most discriminant basis for both reference databases by using Discrete Wavelet Packet Decomposition (DWPD) and Best Discrimination Basis Algorithm (BDBA).

3. *Most Discriminant Basis (MDB 1 and MDB 2)* are bases with maximum discriminant power with regard to each of the two reference databases as determined by a relative entropy discriminant measure.

4. *Theory T_1 and Theory T_2* contain the domain knowledge about features in the MDB's which are interpretable in terms of symbolic target/sound features, about target/sound classes, and about relations between these features and target/sound classes.

5. *Model Construction* creates a model for a given domain theory.

6. *Model M_1 and Model M_2* represent features selected from both sensor data, operations on these features, and relations among these features.

7. *Theory Fusion Operation* fuses domain theories: the theory T_1 and the theory T_2.

8. *Model Fusion Operation* fuses the model M_1 and the model M_2 into one fused model M_f.

9. *Fused Model M_f* represents the fused features.

10. *System Implementation* implements Automatic Multisensor Feature-based Recognition System (AMFRS) using the knowledge of the fused model M_f and the reference databases for training.

11. *AMFRS* is the Automatic Multisensor Feature-based Recognition System using the proposed model-theory based wavelet-features selection (fusion) methodology.

The diagram in Fig. 28 shows that the designer of the AMFRS is given two reference databases of target/sound signatures corresponding to two sensors. The Discrete Wavelet Packet Decomposition (DWPD) is used to transform these signatures into a time/frequency (wavelet) domain and the Best Discrimination Basis Algorithm (BDBA) is used to select the most discriminant basis for each reference database. The designer also knows domain theories that contain the knowledge about interpretable features, target classes and relations between these features and classes. Knowing the most discriminant bases and the domain theories for each sensor, he constructs the models for these domain theories. The knowledge of the fused theory allows the designer to select the right combination of features. The fused theory T_f is derived by using a theory fusion operator to fuse the domain theories T_1 and T_2. If there is no physical law known and therefore there is no theory fusion operator known for fusing data from these two sensors, the designer derives a theory fusion operator manually from a symbolic knowledge about the domain and a relation between the symbolic knowledge and interpretable features. Using the fused theory, he derives manually a model fusion operator. By applying the model fusion operator to the model M_1 and the model M_2, the designer derives the fused model M_f.

In the presented AMFRS methodology for feature selection (fusion), fusion operators are derived by using reduction, expansion and union operators [2]. These reduction, expansion and union operators are defined as follows:

- **Reduction Operator**
  A language L^r is a reduction of the language L if the language L can be written as

$$L = L^r \cup X, \tag{48}$$

  where X is the set of symbols not included in L^r.

We form a model M^r for the language L^r as a reduction of the model $M =< A, I >$ for the language L by restricting the interpretation function $I = I^r \cup I_x$ on $L = L^r \cup X$ to I^r. Therefore

$$M^r =< A, I^r > \tag{49}$$

We form a theory (subtheory) T^r for the language L^r as a reduction of the theory T for the language L by removing sentences from the theory T which are not legal sentences of the language L^r.

- **Expansion Operator**
  A language L^e is an expansion of the language L if the language L can be written as

  $$L^e = L \cup X, \tag{50}$$

  where X is the set of symbols not included in L.

  We form a model M^e for the language $L^e = L \cup X$ as an expansion of the model $M =< A, I >$ for the language L by giving appropriate interpretation I_x to symbols in X. Therefore

  $$M^e =< A, I \cup I_x > \tag{51}$$

  We form a theory T^e for the language L^e as an expansion of the theory T for the language L by adding a set of new legal sentences of the language L^e to the theory T.

- **Union Operator**
  A language L is a union of the languages L_1 and L_2 if the language L can be written as

  $$L = L_1 \cup L_2. \tag{52}$$

  A model $M =< A; R, G, x; I >$ for the language L is a union of the model $M_1 =< A_1; R_1, G_1, X_1; I_1 >$ for the language L_1 and the model $M_2 =< A_2; R_2, G_2, x_2; I_2 >$ for the language L_2 if the model M can be written as

  $$M = M_1 \cup M_2 =< A_1 \cup A_2; R_1 \cup R_2, G_1 \cup G_2, x_1 \cup x_2; I_1 \cup I_2 >, \tag{53}$$

  where R, R_1, R_2 are relations; G, G_1, G_2 are functions; x, x_1, x_2 are constants; and I, I_1, I_2 are interpretation functions.

  We form a theory T for the language L by applying a union operator to the theory T_1 for the language L_1 and to the theory T_2 for the language L_2. Therefore

  $$T = T_1 \cup T_2 \tag{54}$$

6..2 Automatic Multisensor Feature-based Recognition System

Fig. 29 shows a conceptual view of the Automatic Multisensor Feature-based Recognition System (AMFRS) for two sensors.

The meaning of each block in the AMFRS is the following:

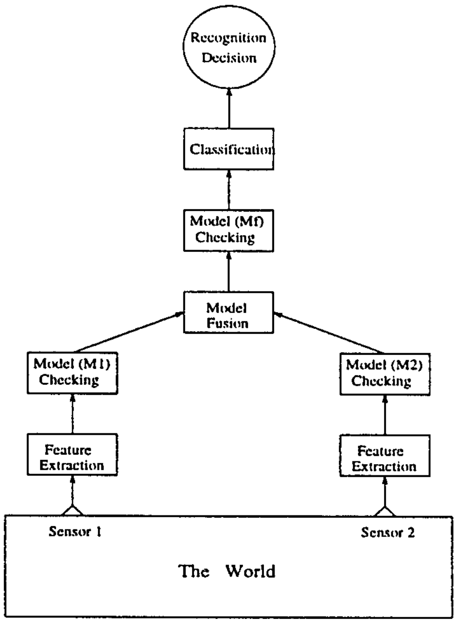

Figure 29: Automatic Multisensor Feature-based Recognition System (AMFRS)

1. *The World* is a domain of the recognition problem with *Sensor1* and *Sensor2* providing information about the domain to the AMFRS.

2. *Feature Extraction* extracts features from measurement data using the BDBA and a model theory framework.

3. *Model (M_1) Checking* tests whether the measurement data represent a model M_1 of a given theory T_1 of the domain.

4. *Model (M_2) Checking* tests whether the measurement data represent a model M_2 of a given theory T_2 of the domain.

5. *Model Fusion* - the fusion operators for the models M_1 and M_2.

6. *Model (M_f) Checking* tests whether the fused data represent a model M_f of a given theory T_f of the domain.

7. *Classification* uses the theory T_f to arrive with a final recognition decision.

There are three types of information available to the AMFRS:

- the target signatures $s_2 \in S$, $i = 1, \cdots, K$, where S is a signal space;
- classes $c_j \in C$, $j = 1, 2$, of targets to be recognized, where C is a target class space;
- symbolic features $b_j \in B$ for each class $c_j \in C$ of targets, where B is a set of symbolic features.

The goal of the AMFRS is to increase the recognition accuracy by using all three types of information in an interpretable way. The *recognition accuracy* is defined as

$$recognition\ accuracy(\%) = \frac{number\ of\ correct\ recognition\ decisions}{total\ number\ of\ recognition\ decisions} * 100 \quad (55)$$

where the *total number of recognition decisions* is equal to the number of target signatures in a test set. Similarly, we define *misclassification rate* as

$$misclassification\ rate(\%) = \frac{number\ of\ incorrect\ recognition\ decisions}{total\ number\ of\ recognition\ decisions} * 100 = 100 - recognition\ accura \quad (56)$$

In this work, we investigate a method for extracting features $f_2 \in F$ of the signal $s_2 \in S$ represented in the wavelet domain, such that these features are interpretable in terms of the symbolic features $b_j \in B$ for every target class $c_j \in C$. We define a recognition problem as a mapping $S \rightarrow C$. This mapping is composed of three mappings: (i) $S \rightarrow W$, (ii) $W \rightarrow F$, and (iii) $F \rightarrow C$. The mapping $S \rightarrow W$ is determined by the Discrete Wavelet Packet Decomposition (DWPD). The DWPD transforms a sensor signal $s_2 \in S$ into a time/frequency (wavelet) representation $W_2 \in W$. The mapping $W \rightarrow F$ is determined by the Best Discrimination Basis Algorithm (BDBA) and a model-theory framework. The BDBA selects the most discriminant basis. The model checker selects features from the most discriminant basis. Conceptually, a mapping $F \rightarrow C$ is determined by a domain theory. The domain theory is an optimal classifier for deterministic target signatures. To apply the domain theory to noisy target signatures, we implement it by using a backpropagation neural network.

6..3 Target Recognition Scenario

As the example of a target recognition problem, we consider a version of a waveform recognition problem, originally examined and applied to a ship recognition project in [1] and then analyzed in [12]. The goal of this problem is to recognize the class to which a given target signature belongs, knowing a reference database of target signatures. The reference database is synthesized using different combinations of three basic waveforms to form three classes of target signatures.

Our formulation of the recognition problem, which is described below, differs from the original waveform recognition problem in the following:

- the number of the measurements in the target signatures is increased from 21 in [1] and 32 in [12] to 128;
- one additional class of target signatures is added, so that we have a four-class problem instead of a three-class problem;
- symbolic features characteristic for each class of targets are assumed to be known;
- the recognition problem is extended to a multisensor scenario by using two reference databases of target signatures (one for each sensor).

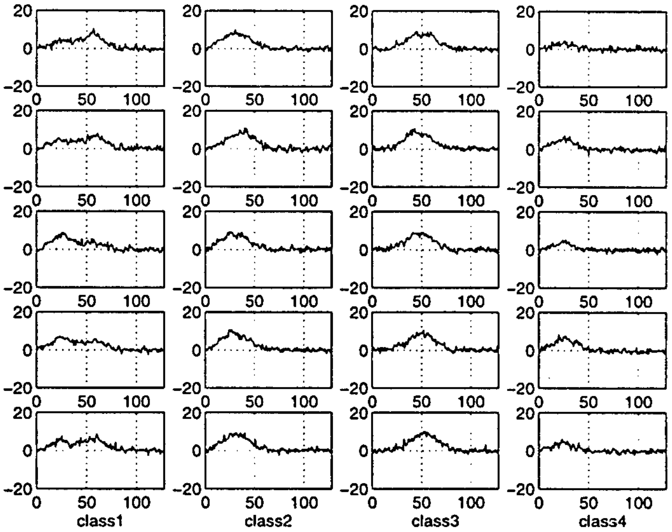

Figure 30: Five Target Signatures for Each of the Four Classes of Targets in DT_1

The DT_1 reference database of target signatures (corresponding to $Sensor_1$) was synthesized using the waveforms

$$h_1(i) = max((24 - |i - 25|)/2, 0), \tag{57}$$

$$h_2(i) = h_1(i - 32), \tag{58}$$

$$h_3(i) = h_1(i - 16), \tag{59}$$

to generate the following four classes of target signatures:

$$class1 = c_1(i) = uh_1(i) + (1 - u)h_2(i) + \epsilon(i), \tag{60}$$

$$class2 = c_2(i) = uh_1(i) + (1 - u)h_3(i) + \epsilon(i), \tag{61}$$

$$class3 = c_3(i) = uh_2(i) + (1 - u)h_3(i) + \epsilon(i), \tag{62}$$

$$class4 = c_4(i) = (1 - u)h_1(i) + \epsilon(i), \tag{63}$$

where $i = 1, \cdots, 128$, u is a uniform random variable and $\epsilon(i)$ are normally distributed random variables. Fig. 30 shows five examples of target signatures for each of the four classes of targets in DT_1.

The DT_2 reference database of target signatures (corresponding to $Sensor_2$) was synthesized using the following four classes of target signatures:

$$class1 = c_1^2(i) = \begin{cases} \epsilon(i) & \text{for } i = 1, \cdots, 10 \\ 2uc_1(i + 49) & \text{for } i = 11, \cdots, 69 \\ 2(1 - u)c_1(i - 69) & \text{for } i = 70, \cdots, 128, \end{cases} \tag{64}$$

$$class2 = c_2^2(i) = \begin{cases} \epsilon(i) & \text{for } i = 1, \cdots, 10 \\ 2uc_2(i + 49) & \text{for } i = 11, \cdots, 69 \\ 2(1 - u)c_2(i - 69) & \text{for } i = 70, \cdots, 128, \end{cases} \tag{65}$$

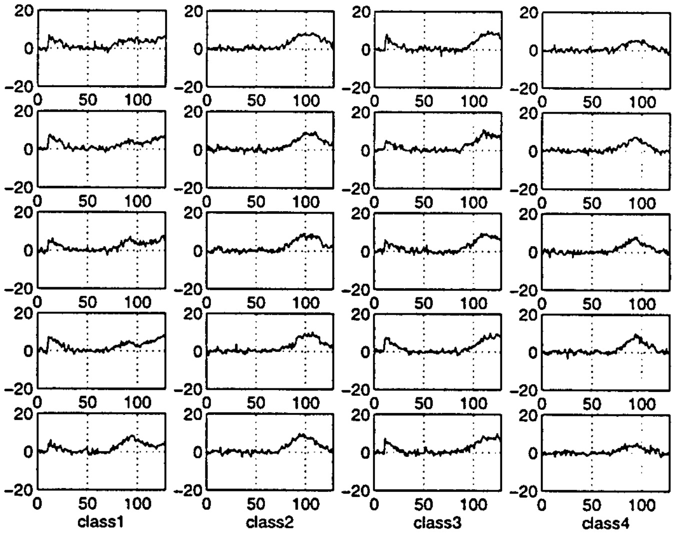

Figure 31: Five Target Signatures for Each of the Four Classes of Targets in DT_2

$$class3 = c_3^2(i) = \begin{cases} \epsilon(i) & \text{for } i = 1, \cdots, 10 \\ 2uc_3(i+49) & \text{for } i = 11, \cdots, 69 \\ 2(1-u)c_3(i-69) & \text{for } i = 70, \cdots, 128, \end{cases} \tag{66}$$

$$class4 = c_4^2(i) = \begin{cases} \epsilon(i) & \text{for } i = 1, \cdots, 10 \\ 2uc_4(i+49) & \text{for } i = 11, \cdots, 69 \\ 2(1-u)c_4(i-69) & \text{for } i = 70, \cdots, 128, \end{cases} \tag{67}$$

where $i = 1, \cdots, 128$, u is a uniform random variable and $\epsilon(i)$ are normally distributed random variables. Fig. 31 shows five examples of target signatures in DT_2 for each of the four classes of targets.

6..4 Feature Selection for Target Recognition

Each of the target signatures in the DT_1 reference database is transformed into the wavelet domain using the DWPD with the Daubechies_6 compactly supported wavelets with extreme phase and highest number of vanishing moments compatible with their support width. A complete orthonormal Most Discriminant Basis (MDB) is selected using the BDBA. The total number of components (wavelet coefficients) in this basis is equal to 128, i.e., it is equal to the number of samples in the target signature. Fig. 32 shows the MDB for the DT_1 reference database and the discriminant value of each component as determined by a relative entropy measure. The MDB consists of the two subbases on the third level of the DWPD decomposition and one subbasis on the second level of the decomposition. In order to limit the computational complexity of recognition, we apply a domain theory (symbolic knowledge about the target signatures) for the DT_1 reference database to select $n_1 < 128$ wavelet coefficients (features) from the complete MDB. Our aim is to select n_1 features that, in addition to significant discriminant power, are interpretable in terms of the following symbolic target signature features:

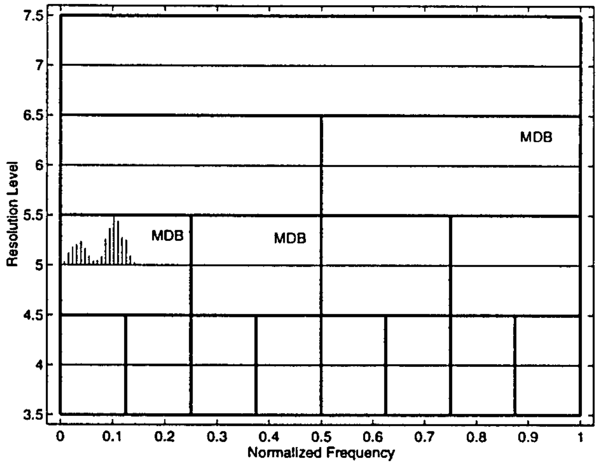

Figure 32: Most Discriminant Basis (MDB) for the DT_1 Reference Database

- rising edges, where the value of the signal is increasing within a given time interval;
- falling edges, where the value of the signal is decreasing within a given time interval;
- zero edges, where the value of the signal remains around zero within a given time interval;
- peaks, where the value of the signal has its local maximum.

The value of selected features indicates which one of the four symbolic features described above is present in the target signature, while the position of selected features in the MDB indicates the time interval within which a given symbolic feature is present in the target signature.

Below, we describe an appropriate language, a model, and a theory for $Sensor_1$. The symbol $\leq$ is a *logical symbol* with the usual interpretation as a linear order relation on the universe.

1. **Language L_1**
   A feature-level language L_1 for $Sensor_1$ is the following:

$$L_1 = \{class1_1, class2_1, class3_1, class4_1, f_1, 0, 1, C_{r_0}, C_{r_1}, \ldots, C_{r_4}, C_{r_5}\}, \qquad (68)$$

where

- $class1_1, class2_1, class3_1, class4_1$ are 6-placed relation symbols (classes of targets to be recognized using $Sensor_1$),
- f_1 is an 1-placed function symbol (a function that maps features' indices into features' values),
- $0, 1$ are constant symbols,
- $C_{r_0}, C_{r_1}, \ldots, C_{r_4}, C_{r_5}$ are constant symbols (features' indices).

2. **Theory T_1**
A feature-level theory T_1 for $Sensor_1$ is the following.

$$\begin{aligned} class1_1(y_0 \dots y_5) &\iff (\forall_{y_0,\dots,y_5})\ ((y_0 = f_1(C_{r_0}) \wedge\ \dots\ \wedge y_5 = f_1(C_{r_5})) \Rightarrow \\ &\quad (1 \le y_0 \le y_2 \le y_1\ \wedge\ 1 \le y_3 \le y_5 \le y_4)), \quad (69)\\ class2_1(y_0 \dots y_5) &\iff (\forall_{y_0,\dots,y_5})\ ((y_0 = f_1(C_{r_0}) \wedge\ \dots\ \wedge y_5 = f_1(C_{r_5})) \Rightarrow \\ &\quad (1 \le y_0 \le y_1\ \wedge\ y_0 \le y_2\ \wedge\ 1 \le y_5 \le y_4 \le y_3)), \quad (70)\\ class3_1(y_0 \dots y_5) &\iff (\forall_{y_0,\dots,y_5})\ ((y_0 = f_1(C_{r_0}) \wedge\ \dots\ \wedge y_5 = f_1(C_{r_5})) \Rightarrow \\ &\quad (0 = y_0 \le y_1 \le y_2\ \wedge\ 1 \le y_5 \le y_4 \le y_3)), \quad (71)\\ class4_1(y_0 \dots y_5) &\iff (\forall_{y_0,\dots,y_5})\ ((y_0 = f_1(C_{r_0}) \wedge\ \dots\ \wedge y_5 = f_1(C_{r_5})) \Rightarrow \\ &\quad (1 \le y_0 \le y_2 \le y_1\ \wedge\ y_3 = y_4 = y_5 = 0)), \quad (72) \end{aligned}$$

$$C_{r_0} \le C_{r_1} \le C_{r_2} \le C_{r_3} \le C_{r_4} \le C_{r_5}. \quad (73)$$

3. **Model M_1**
A feature-level model M_1 of the theory T_1 is defined as

$$M_1 =< A_1, I_1 >, \quad (74)$$

or as

$$M_1 =< A_1; class1_1, class2_1, class3_1, class4_1, f_1, 0, 1, \dots, 5; I_1 >, \quad (75)$$

where

- $A_1 = \{0, \dots, 5\}$ is a universe of the model M_1,
- $class1_1, class2_1, class3_1, class4_1 \subset A_1^6$ are relations,
- $f_1 : A_1 \to A_1$ is a function,
- I_1 is an interpretation function that maps symbols of the language L_1 to appropriate relations, functions and constants in the universe A_1. It assigns relations $class1_1 \subset A_1^6$, $class2_1 \subset A_1^6$, $class3_1 \subset A_1^6$, and $class4_1 \subset A_1^6$ in the model M_1 to symbols $class1_1$, $class2_1$, $class3_1$, and $class4_1$ in the language L_1 respectively. We use $class1_1$, $class2_1$, $class3_1$, and $class4_1$ to denote both the relation symbols in L_1 and the relations in M_1. I_1 also assigns a function $f_1 : A_1 \to A_1$ in the model M_1 to a symbol f_1 in the language L_1. We use f_1 to denote both the function symbol in L_1 and the function in M_1. And finally, I_1 assigns constants $0, \dots, 5$ in the model M_1 to the constant symbols $C_{r_0}, \dots, C_{r_5}$ in the language L_1 respectively and special constants $0, 1$ in the model M_1 to the constant symbols $0, 1$ in the language L_1.

The $n_1 = 6$ selected features from the most discriminant basis (MDB) using the domain theory are shown in Fig. 33. Fig. 34 shows the relationship between the selected features and symbolic target features in ideal, not noisy conditions. All six features, depending on their value and position, carry the information about the type of symbolic target features and their position in a target signature. The first three features correspond to symbolic features in the beginning part of a target signature, while the last three features correspond to symbolic features in the middle part of a target signature. In particular:

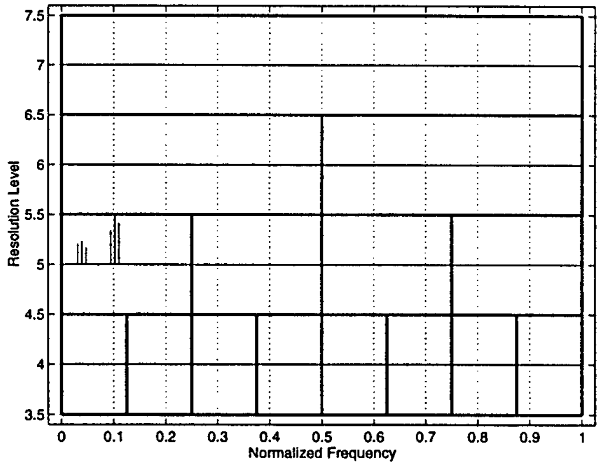

Figure 33: Features Selected From the MDB for the DT_1 Database Using the Domain Theory

- the value of the first feature indicates the existence of a zero edge characteristic for target signatures belonging to *class*3 or the existence of a rising edge characteristic for target signatures belonging to *class*1, *class*2, and *class*4 (see Fig. 34);
- the value of the second feature indicates the existence of a peak characteristic for target signatures belonging to *class*1 or the existence of a rising edge characteristic for target signatures belonging to *class*2, *class*3 and *class*4 (see Fig. 34);
- the value of the third feature indicates the existence of a rising edge characteristic for target signatures belonging to *class*2 and *class*3 or the existence of a falling edge characteristic for target signatures belonging to *class*1 and *class*4 (see Fig. 34);
- the value of the fourth feature indicates the existence of a rising edge characteristic for target signatures belonging to *class*1, the existence of a zero edge characteristic for target signatures belonging to *class*4, or the existence of a falling edge characteristic for target signatures belonging to *class*2 and *class*3 (see Fig. 34);
- the value of the fifth feature indicates the existence of a peak characteristic for target signatures belonging to *class*1, the existence of a zero edge characteristic for target signatures belonging to *class*4, or the existence of a falling edge characteristic for target signatures belonging to *class*2 and *class*3 (see Fig. 34);
- the value of the sixth feature indicates the existence of a falling edge characteristic for target signatures belonging to *class*1, *class*2 and *class*3, or the existence of a zero edge characteristic for target signatures belonging to *class*4 (see Fig. 34).

The features selected from the MDB using the theory T_1 are different from the Most Discriminant Wavelet Coefficients (MDWC) which are shown in Fig. 35.

In the same manner as for the $Sensor_1$, we define an appropriate language, a model, and a theory for $Sensor_2$. The symbol $\leq$ is again assumed to be a *logical symbol* with the usual interpretation as a linear order relation on the universe.

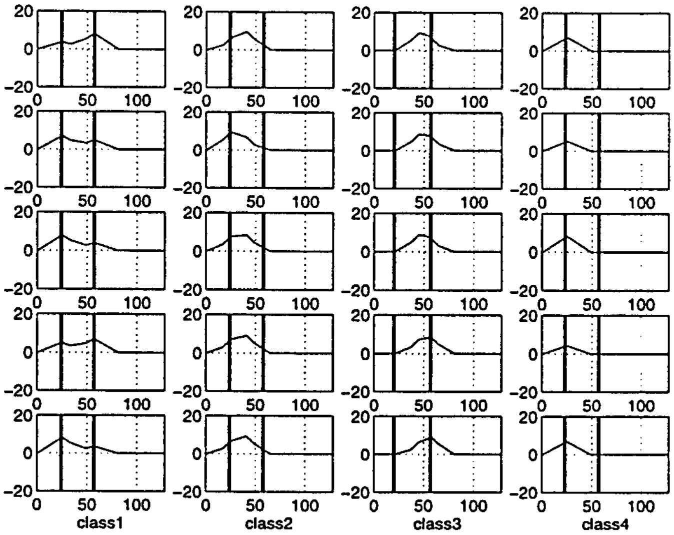

Figure 34: Relationship Between the Selected Features and Symbolic Target Features for the DT_1 Database

1. **Language L_2**
   A feature-level language L_2 for $Sensor_2$ is the following:

$$L_2 = \{class1_2, class2_2, class3_2, class4_2, f_2, 0, 1, 2, C_{i_0}, C_{i_1}, \ldots, C_{i_7}, C_{i_8}\}, \tag{76}$$

   where

   - $class1_2, class2_2, class3_2, class4_2$ are 9-placed relation symbols (classes of targets to be recognized using $Sensor_2$),
   - f_2 is an 1-placed function symbol (a function that maps features' indices into features' values),
   - $0, 1, 2$ are constant symbols,
   - $C_{i_0}, C_{i_1}, \ldots, C_{i_7}, C_{i_8}$ are constant symbols (features' indices).

2. **Theory T_2**
   A feature-level theory T_2 for $Sensor_2$ is the following.

$$\begin{aligned} class1_2(y_0 \ldots y_8) \iff & (\forall_{y_0, \ldots, y_8})\ ((y_0 = f_2(C_{i_0}) \wedge \ \ldots \ \wedge y_8 = f_2(C_{i_8})) \Rightarrow \\ & (2 \le y_0 \le y_2 \le y_1 \ \wedge \ 2 \le y_5 \le y_3 \le y_4 \ \wedge \\ & 2 \le y_6 \le y_7 \ \wedge \ 2 \le y_8 \le y_7)), \end{aligned} \tag{77}$$

$$\begin{aligned} class2_2(y_0 \ldots y_8) \iff & (\forall_{y_0, \ldots, y_8})\ ((y_0 = f_2(C_{i_0}) \wedge \ \ldots \ \wedge y_8 = f_2(C_{i_8})) \Rightarrow \\ & (1 \le y_2 \le y_1 \le y_0 \ \wedge \ 2 \le y_3 \le y_4 \le y_5 \ \wedge \\ & y_7 \le y_6 \le 1 \ \wedge \ y_8 = 0)), \end{aligned} \tag{78}$$

$$\begin{aligned} class3_2(y_0 \ldots y_8) \iff & (\forall_{y_0, \ldots, y_8})\ ((y_0 = f_2(C_{i_0}) \wedge \ \ldots \ \wedge y_8 = f_2(C_{i_8})) \Rightarrow \\ & (2 \le y_0 = y_1 \ \wedge \ y_2 \le y_1 \ \wedge \ 1 \le y_3 \le y_4 \le y_5 \ \wedge \end{aligned}$$

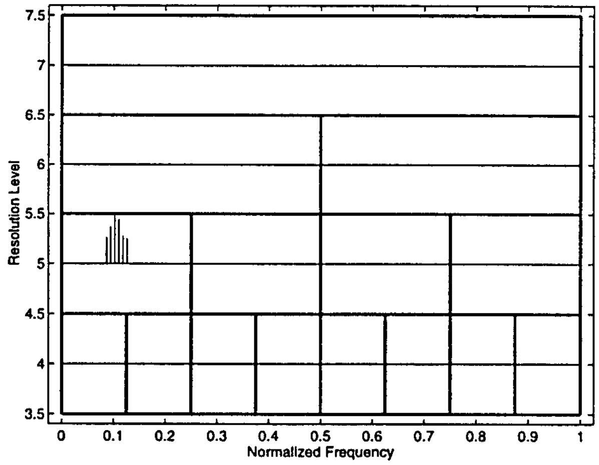

Figure 35: Most Discriminant Wavelet Coefficients (MDWC) Selected from the MDB for the DT_1 Database as Features

$$2 \leq y_6 \leq y_7 \ \wedge \ 2 \leq y_8 \leq y_7)), \tag{79}$$

$$\begin{aligned} class4_2(y_0 \ldots y_8) \iff & (\forall_{y_0,\ldots,y_8}) \ ((y_0 = f_2(C_{i_0}) \wedge \ \ldots \ \wedge y_8 = f_2(C_{i_8})) \Rightarrow \\ & (y_0 = y_1 = y_2 = y_6 = y_7 = y_8 = 0 \ \wedge \ 2 \leq y_3 \leq y_4 \ \wedge \\ & 2 \leq y_5 \leq y_4)), \end{aligned} \tag{80}$$

$$C_{i_0} \leq C_{i_1} \leq C_{i_2} \leq C_{i_3} \leq C_{i_4} \leq C_{i_5} \leq C_{i_6} \leq C_{i_7} \leq C_{i_8}. \tag{81}$$

3. **Model M_2**

A feature-level model M_2 of the theory T_2 is defined as

$$M_2 =< A_2, I_2 >, \tag{82}$$

or as

$$M_2 =< A_2; class1_2, class2_2, class3_2, class4_2, f_2, 0, 1, \ldots, 8; I_2 >, \tag{83}$$

where

- $A_2 = \{0, \ldots, 8\}$ is a universe of the model M_2,
- $class1_2, class2_2, class3_2, class4_2 \subset A_2^9$ are relations,
- $f_2 : \ A_2 \rightarrow A_2$ is a function,
- I_2 is an interpretation function that maps symbols of the language L_2 to appropriate relations, functions and constants in the universe A_2. It assigns relations $class1_2 \subset A_2^9$, $class2_2 \subset A_2^9$, $class3_2 \subset A_2^9$, and $class4_2 \subset A_2^9$ in the model M_2 to symbols $class1_2$, $class2_2$, $class3_2$, and $class4_2$ in the language L_2 respectively. We use $class1_2$, $class2_2$, $class3_2$, $class4_2$ to denote both the relation symbols in L_2 and the relations in M_2. I_2 also assigns a function $f_2 : \ A_2 \rightarrow A_2$ in the model M_2 to a symbol f_2 in the language L_2. We use f_2 to denote both the function symbol in L_2 and the function in M_2. And finally, I_2 assigns constants $0, \ldots, 8$ in the model M_2 to the constant symbols $C_{i_0}, \ldots, C_{i_8}$ in the language L_2 respectively and special constants $0, 1, 2$ in the model M_2 to the constant symbols $0, 1, 2$ in the language L_2.

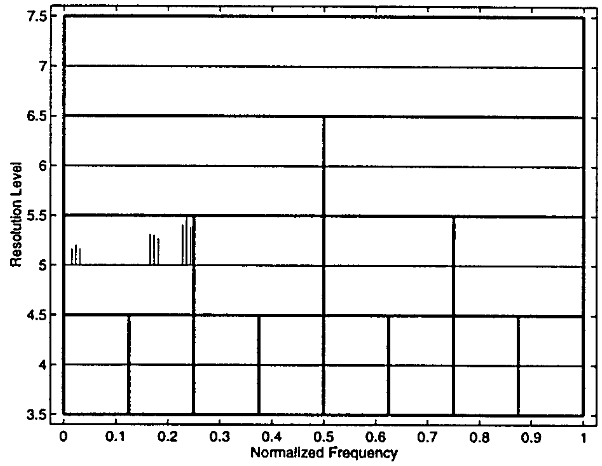

Figure 36: Features Selected From the MDB for the DT_2 Database Using the Domain Theory

The $n_2 = 9$ selected features from the MDB using the domain theory (symbolic knowledge about the target signatures in the DT_2 database) are shown in Fig. 36. More details on feature selection for $Sensor_2$ can be found in [9].

1. **Fused Language L_f**
   A feature-level fused language L_f is the following:

$$L_f = \{class1, class2, class3, class4, f_f, 0, 1, 2, C_{f_0}, C_{f_1}, \ldots, C_{f_7}, C_{f_8}\}, \tag{84}$$

   where

   - $class1, class2, class3, class4$ are 9-placed relation symbols (classes of targets to be recognized using fused data),
   - f_f is an 1-placed function symbol (a function that maps features' indices into features' values),
   - $0, 1, 2$ are constant symbols,
   - $C_{f_0}, C_{f_1}, \ldots, C_{f_7}, C_{f_8}$ are constant symbols (features' indices).

2. **Fused Theory T_f**
   A feature-level fused theory T_f is the following.

$$\begin{aligned} class1(y_0 \ldots y_8) \iff & (\forall_{y_0,\ldots,y_8})\ ((y_0 = f_f(C_{f_0}) \wedge \ \ldots\ \wedge y_8 = f_f(C_{f_8})) \Rightarrow \\ & (1 \leq y_0 \leq y_2 \leq y_1\ \wedge\ 1 \leq y_3 \leq y_5 \leq y_4\ \wedge \\ & 2 \leq y_6 \leq y_7\ \wedge\ 2 \leq y_8 \leq y_7)), \end{aligned} \tag{85}$$

$$\begin{aligned} class2(y_0 \ldots y_8) \iff & (\forall_{y_0,\ldots,y_8})\ ((y_0 = f_f(C_{f_0}) \wedge \ \ldots\ \wedge y_8 = f_f(C_{f_8})) \Rightarrow \\ & (1 \leq y_0 \leq y_1\ \wedge\ y_0 \leq y_2\ \wedge\ 1 \leq y_5 \leq y_4 \leq y_3\ \wedge \\ & y_7 \leq y_6 \leq 1\ \wedge\ y_8 = 0)), \end{aligned} \tag{86}$$

$$class3(y_0 \ldots y_8) \iff (\forall_{y_0,\ldots,y_8})\ ((y_0 = f_f(C_{f_0}) \wedge \ \ldots\ \wedge y_8 = f_f(C_{f_8})) \Rightarrow$$

$$\begin{aligned} & (0 = y_0 \leq y_1 \leq y_2 \ \wedge \ 1 \leq y_5 \leq y_4 \leq y_3 \ \wedge \\ & 2 \leq y_6 \leq y_7 \ \wedge \ 2 \leq y_8 \leq y_7)), \end{aligned} \tag{87}$$

$$\begin{aligned} class4(y_0 \ldots y_8) \iff & (\forall_{y_0, \ldots, y_8}) \ ((y_0 = f_f(C_{f_0}) \wedge \ \ldots \ \wedge y_8 = f_f(C_{f_8})) \Rightarrow \\ & (1 \leq y_0 \leq y_2 \leq y_1 \ \wedge \ y_3 = y_4 = y_5 = y_6 = y_7 = y_8 = 0)), \end{aligned} \tag{88}$$

$$C_{f_0} \leq C_{f_1} \leq C_{f_2} \leq C_{f_3} \leq C_{f_4} \leq C_{f_5} \leq C_{f_6} \leq C_{f_7} \leq C_{f_8}, \tag{89}$$

where

$$\begin{aligned} & C_{f_0} = C_{r_0} \wedge C_{f_1} = C_{r_1} \wedge C_{f_2} = C_{r_2} \wedge C_{f_3} = C_{r_3} \wedge C_{f_4} = C_{r_4} \wedge \\ & C_{f_5} = C_{r_5} \wedge C_{f_6} = C_{i_6} \wedge C_{f_7} = C_{i_7} \wedge C_{f_8} = C_{i_8}, \end{aligned} \tag{90}$$

$$\begin{aligned} & f_f(C_{f_0}) = f_1(C_{r_0}) \wedge f_f(C_{f_1}) = f_1(C_{r_1}) \wedge f_f(C_{f_2}) = f_1(C_{r_2}) \wedge \\ & f_f(C_{f_3}) = (f_1(C_{r_3}) \wedge f_f(C_{f_4}) = f_1(C_{r_4}) \wedge f_f(C_{f_5}) = f_1(C_{r_5}) \wedge \\ & f_f(C_{f_6}) = f_2(C_{i_6}) \wedge f_f(C_{f_7}) = f_2(C_{i_7}) \wedge f_f(C_{f_8}) = f_2(C_{i_8}). \end{aligned} \tag{91}$$

3. **Fused Model M_f**

A feature-level fused model M_f of the theory T_f is defined as

$$M_f =< A, I_f >, \tag{92}$$

or as

$$M_f =< A; class1, class2, class3, class4, f_f, 0, 1, \ldots, 8; I_f >, \tag{93}$$

where

- $A = \{0, \ldots, 8\}$ is a universe of the model M_f,
- $class1 \subset A^9$ is a relation, where $class1 = class1'_2 \times class1_1$ and $class1'_2 = class1_2 \cap (A \cap \{6,7,8\})^3$,
- $class2 \subset A^9$ is a relation, where $class2 = class2_1 \times class2'_2$ and $class2'_2 = class2_2 \cap (A \cap \{6,7,8\})^3$,
- $class3 \subset A^9$ is a relation, where $class3 = class3_1 \times class3'_2$ and $class3'_2 = class3_2 \cap (A \cap \{6,7,8\})^3$,
- $class4 \subset A^9$ is a relation, where $class4 = class4_1 \times class4'_2$ and $class4'_2 = class4_2 \cap (A \cap \{6,7,8\})^3$,
- $f_f : \ A \rightarrow A$ is a function, where $f_f = f_1 \cup f'_2$ and $f'_2 = f_2 \mid_{A \cap \{6,7,8\}}$,
- I_f is an interpretation function that maps symbols of the language L_f to appropriate relations, functions and constants in the universe A. It assigns relations $class1 \subset A^9$, $class2 \subset A^9$, $class3 \subset A^9$, and $class4 \subset A^9$ in the model M_f to symbols $class1$, $class2$, $class3$, and $class4$ in the language L_f respectively. We use $class1$, $class2$, $class3$, $class4$ to denote both the relation symbols in L_f and the relations in M_f. I_f also assigns a function $f_f : \ A \rightarrow A$ in the model M_f to a symbol f_f in the language L_f. We use f_f to denote both the function symbol in L_f and the function in M_f. And finally, I_f assigns constants $0, \ldots, 8$ in the model M_f to the constant symbols $C_{f_0}, \ldots, C_{f_8}$ in the language L_f respectively and special constants $0, 1, 2$ in the model M_f to the constant symbols $0, 1, 2$ in the language L_f.

The fused theory and fused model are used to fuse the feature vector consisting of $n_1 = 6$ features from the DT_1 database and the feature vector consisting of $n_2 = 9$ features from the DT_2 database into one fused feature vector consisting of $n_f = 9$ interpretable features. In particular:

- the first three fused features of this fused feature vector are the first three features selected for $Sensor_1$, whose value indicates the existence of the first peak characteristic for target signatures belonging to $class1$ and $class4$, the existence of a rising edge characteristic for target signatures belonging to $class2$, or the existence of a zero edge characteristic for target signatures belonging to $class3$;

- the next three fused features are the last three features selected for $Sensor_1$, whose value indicates the existence of the second peak characteristic for target signatures belonging to $class1$, the existence of a falling edge characteristic for target signatures belonging to $class2$ and $class3$, or the existence of a zero edge characteristic for target signatures belonging to $class4$;

- the last three fused features of this fused feature vector are the first three features selected for $Sensor_2$, whose value indicates the existence of a peak characteristic for target signatures belonging to $class1$ and $class3$, or the existence of a zero edge characteristic for target signatures belonging to $class2$ and $class4$.

6..5 Results of Experiments

We test the recognition accuracy of the AMFRS by using two test databases with target signatures corresponding to the DT_1 and DT_2 reference databases. Each of the target signatures in these databases is transformed into the wavelet domain using the DWPD. Then, we select one target signature from each of the test databases in such a way that these two target signatures correspond to the same target. A framework of model theory (SDFS) is used to extract features from each of these two target signatures, and to fuse them into one combined feature vector. This feature vector is used as an input vector into a backpropagation neural network. The backpropagation neural network arrives with a final recognition decision.

Fig. 37 shows the resulting misclassification rates for different levels of noise for the multisensor (using the DT_1 and DT_2 target signatures) target recognition problem. This figure also shows the misclassification rate of the target recognition system using Most Discriminant Wavelet Coefficients (MDWC) as features. The misclassification results show that the AMFRS has a better recognition accuracy than a MDWC based target recognition system.

6..6 Conclusions and Future Research

A multisensor recognition methodology (AMFRS) is proposed based on features selected by utilizing the Best Discrimination Basis Algorithm (BDBA), symbolic knowledge about the domain, and a model-theory based fusion framework. The symbolic knowledge about the domain is incorporated into the AMFRS through a domain theory using a model-theory based fusion framework.

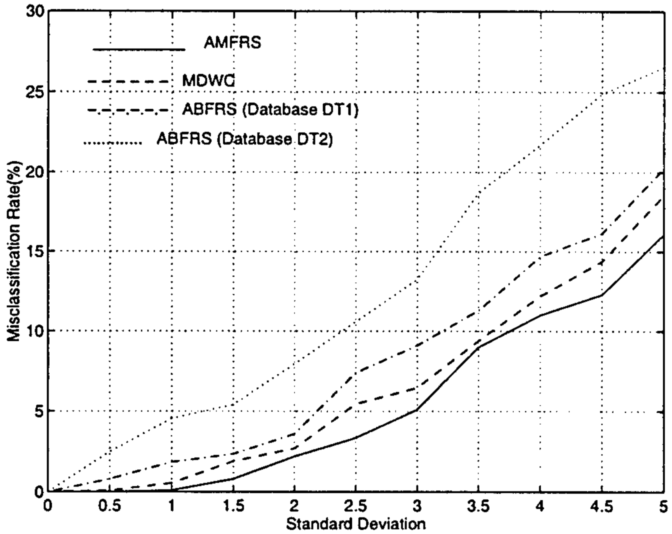

Figure 37: Misclassification Rates for the Multisensor Target Recognition Using Model Theory (AMFRS) and Most Discriminant Wavelet Coefficients (MDWC) for Feature Selection

A model of a domain theory represents a bridge between a symbolic knowledge about the domain and measurement data.

The simulation results show that the AMFRS has a better recognition accuracy than a MDWC based recognition system with respect to the test databases. The MDWC based target recognition system adapts well to the training set of target signatures, but it does not have enough generalization power to perform equally well with the test data. The MDWC used to train a classifier, in the discussed recognition problem, are concentrated only in one time/frequency area. We are able to increase the generalization power of the target recognition system by selecting interpretable features which are more spread across the time/frequency domain and which capture more of symbolic target features. To select such interpretable features, we utilize the symbolic knowledge (fused theory) specific to the recognition problem domain.

In this work, we build the domain theory manually based on the analysis of the target signatures in the reference database (symbolic knowledge) and the most discriminant bases determined by using the BDBA. In the future research we are going to incorporate learning capabilities into the AMFRS.

References

[1] L. Breiman, J. H. Friedman, R. A. Olshen and C. J. Stone, *Classification and Regression Trees.* Belmont, California: Wadsworth International Group, 1984.

[2] C. C. Chang and H. J. Keisler. *Model Theory.* Amsterdam, New York, Oxford, Tokyo: North Holland, 1992.

[3] R. R. Coifman and M. V. Wickerhauser, "Entropy-Based Algorithms for Best Basis Selection," *IEEE Transactions on Information Theory,* vol.38, no.2, pp. 713–718, March 1992.

[4] I. Daubechies, *Ten Lectures on Wavelets.* Philadelphia, Pennsylvania: Society for Industrial and Applied Mathematics, 1992.

[5] D. L. Hall, *Mathematical Techniques in Multisensor Data Fusion.* Boston-London: Artech House, 1992.

[6] N. Immerman, "Languages Which Capture Complexity Classes," in *Proc. 15th Ann. ACM Symp. on the Theory of Computing,* pp. 347–354, 1983.

[7] M.M. Kokar and J. Tomasik, *Towards a Formal Theory of Sensor/Data Fusion.* Progress Report, Northeastern University, May 1994.

[8] Z. Korona and M.M. Kokar, "Multiresolution Multisensor Target Identification," in *Proc. IEEE Int'l Conf. Multisensor Fusion and Integration for Intelligent Systems (MFI'94),* pp. 501–506, Las Vegas, Nevada, Oct. 1994.

[9] Z. Korona, "Model Theory Based Feature Selection for Multisensor Recognition," *Ph.D. Dissertation,* Northeastern University, 1996.

[10] "Wavelet-Based Recognition Using Model Theory for Feature Selection," in *Proc. SPIE Aerospace/Defense Sensing and Controls Symposium: Wavelet Applications III,* pp. 256–266, Orlando, FL, April 1996.

[11] R. C. Luo and M. G. Kay, "Multisensor Integration and Fusion in Intelligent Systems," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 19, no. 5, pp. 901–931, 1989.

[12] N. Saito, "Local Feature Extraction and Its Applications Using a Library of Bases," *Ph.D. Dissertation,* Yale University, Dec. 1994.